1,167,762

PATENT SPECIFICATION

DRAWINGS ATTACHED

1,167,762

Date of Application (No. 46813/66) and filing Complete Specification: 19 October, 1966.

Application made in United States of America (No. 509909) on 26 November, 1965.

Complete Specification Published: 22 October, 1969.

Index at Acceptance:—G4 A (2C, 3W2, 5A, 5B, 6G, 6H, 6M1, 9X, 9C, 9D, 12N, 12P, 13E, 16H).

International Classification:—G 06 f 3/00.

COMPLETE SPECIFICATION

Input-Output Data Service Computer

We, BURROUGHS CORPORATION, a Corporation organised under the laws of the State of Michigan, United States of America, of 6071, Second Avenue, Detroit, State of Michigan, United States of America, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed to be particularly described in and by the following statement:—

This invention relates to an input-output programmable computer for use with a data communication system and is concerned particularly with a programmable computer for the control of input and output devices which are incorporated in electronic data processing systems.

In the operation of electronic data processing systems it is usual for the overall speed of operation of the system to be limited or restricted by relatively slow speed input and output devices connected in the system. This limitation upon system speed has remained despite the use of input and output control units and the use of such techniques as multiplexing of information transfer and scanning of input and output devices requiring service.

The actual computation performed by central processors has customarily been the most rapidly performed operation in data processing systems. The cycle of operation of memory systems, though in the past relatively slow in comparison to arithmetic computation, has been shortened considerably by the development of much higher speed memories such as thin film arrays, tunnel diode and cryogenic memory devices. Even the programming of electronic data processing systems has been improved to enable more rapid programming of systems by the use of machine-oriented programming language and the use of program compilers. It has, therefore, become increasingly desirable to increase the speed of information transfer into and out of system memories and to increase the independence of operation of input and output information transfers to permit central processors to utilize a higher portion of the time available for performance of computations.

Independent operation of input and output information transfers is desired to free central processors from direct step-by-step control of input and output transfer operations — the so-called bookkeeping operations — so that the central processor can utilize more of the time available for performing computations near its maximum rate of operation. In certain modular data processing systems, independent operation of input and output transfer operations is desirable to also permit fully independent operation of the central processor modules with the memory modules.

To improve the speed and independence of input and output information transfers, some past data processing systems have incorporated general purpose computers as satellites to attend to and direct the input and output operations and have used buffering devices between the input and output devices and the satellite computer for increasing the rate of information transfer thereto. The use of such satellite computers for input and output control, however, has not eliminated the restrictions placed on system operating speed and efficiency by the demands of the input and output devices since central processor attention and direction to the input and output operations was still necessary for providing step-by-step control into the satellite's memory.

The buffers connected between input and output devices and the satellite computers

[Price 4s. 6d.]

were also inefficient since the input and output transfers were buffered to equal size words or characters of information, thus requiring more frequent service of some devices than others. The nature of these general purpose satellite computers also limited their usefulness for incorporation into modular data processing systems since central processor time would be necessary for developing and causing delivery of sets of transfer instructions directly into the general purpose satellite computer's memory for directing the satellite computer's control of input and output information transfers.

Accordingly, it is an object of the subject invention to provide a programmable input-output data service computer for controlling information transfers between input and output devices and a central memory via a magnetic disc file if desired, without continual direct control or instruction by a central processor.

According to this invention, an input-output programmable computer for use with a data communication system comprises a processing means for constructing information transfer descriptors under program control, an input-output controller comprising a plurality of input output terminals and a local memory for storing address information relating to locations of input/output channel programs stored in the system and said information transfer descriptors, and a data service means coupled to said local memory for decoding the information transfer descriptors and transferring data between said input/output terminals or between an input output terminal and the system in response to different ones of said descriptors.

This input-output programmable computer may be used, for example, in a data processing system to free the central processor in the data processing system from step-by-step control of input and output information transfers and thus enable a greater computation efficiency by the central processor.

As will be described later, it becomes convenient to deliver and accept different size bytes of information to and from input and output devices rather than to buffer each input and output transfer in equal size words and thus permit the approach to equal frequency of service of each input and output device.

It is possible to store input-output channel controls in the local memory of the input-output computer to enable the computer to access directly program instructions stored in a main memory for constructing channel descriptors for use in directing the performance of information transfers.

The input-output programmable computer may be used in an electronic data communication system having a main memory, and the aforesaid local memory in the input-output programmable computer in this case may form local storage means for storing program address information relating to the main memory and the information transfer descriptors, and said data service means may then serve to transfer data between input and output devices or between an input or output device and the main memory in response to different ones of the descriptors.

In an arrangement where the local storage means stores main memory address information of input-output channel programs, the processing means may be coupled to the local storage means for constructing information transfer descriptors directing input-output data transfers by executing programs stored in main memory which it addresses directly upon obtaining address information from the local storage means. The data service means may then perform information transfers under control of the descriptors constructed by the processing means.

The invention furthermore includes within its scope an input-output programmable computer for use with a data communication system, said computer comprising a processing means for constructing information transfer descriptors under program control and for addressing a stack of input/output data transfer jobs stored in said system to obtain system address information of input/output data transfer programs to be executed, an input-output controller comprising a plurality of input output terminals and a local memory for storing address information relating to locations of input output channel programs stored in the system and said information transfer descriptors, and a data service means coupled to said local memory for decoding the information transfer descriptors and transferring data between said input output terminals or between an input/output terminal and the system in response to different ones of said descriptors.

The aforementioned input-output data service means may provide for scanning input and output channels in order of priority and controlling information transfer in varying size bytes and number of bytes with a memory in accordance with unique channel descriptors available for each input and output channel. There may also be provided an input-output processor unit capable of addressing and executing externally stored instructions under system program control for sequential on-demand construction of channel descriptors for use in the control of input and output information transfers. The input-output processor unit may further include operand field discrimina-

tion means for controlling the field to be operated upon in modifying channel control instruction and parameter address information and for constructing channel descriptors for the control of information transfers.

The input-output programmable computer thus can form an input-output data service computer and will be so referred to where appropriate. This data service computer can be capable of direct access to program instructions and parameters stored in an external main memory in accordance with the state word address containing portion of the channel controls which are stored within its local memory. Each channel state word in local memory may comprise an instruction base address, a parameter base address, an instruction counter, and index registers for use by the input-output processor unit in addressing main memory locations. Main memory address locations can thus be directly addressed for obtaining instruction steps and parameters or operands for use in constructing channel descriptors. This local memory-to-main memory addressing technique for accessing instructions and parameters stored in main memory locations is completely under the control of the input-output data service computer and does not interfere with the operation of the central processor in any system in which it is used, nor does it require continual direction or control by the system central processor for obtaining instructions. An important feature of the separate storage of instructions and parameters in main memory is that the input-output processor unit can utilize a given program set of instructions or program routine for constructing different input-output channel controls. Each channel may have assigned to it a unique set of parameters for use in the execution of these universally available program routines.

The input-output data service computer may process fully jobs assigned to it by a system executive scheduling program which may include a number of program routines. Each new job may provide the input-output processing unit with a new state word for one of its channel controls which identifies a program to be processed and supplies parameters for use in executing the program instructions. Once the input-output computer begins executing the program identified by a job, it will construct a descriptor in the processing unit, perform the transfer described by the descriptor, and return each time to the job program for instructions for constructing new descriptors until the entire input-output job is completely performed. While the data service means is implementing a descriptor for a channel, the processing unit may continue to serve other channels by re-entering the programs in main memory identified by the jobs for each channel for constructing new descriptors therefor as needed.

The input-output programmable computer of the subject invention can be utilized in combination in a data processing system having a main memory such that data transfers between various input and output devices, as well as transfers between input and output devices and main memory, can be effected without any intervention by, or interruption of, the central processor. In particular, transfers between a disc file (or magnetic drum) and other input and output devices can be so effected. For example, information coming from an input device, such as a card reader, can be assembled on the disc until detection of an indication that the assembled information is complete, whereupon the input-output processor will return to the disc channel program for another task. This task may well be that of taking the information once assembled on the disc and transferring it in a block to main memory. In such device-to-device transfers, a buffer area in the main memory is assigned to the input or output channel involved.

In the case of a transfer from an input device to disc file, and subsequently to main memory, the sequence of information transfer is as follows: Under control of a descriptor the input-output data service means receives data words or bytes from the inputting device when it has priority, accumulates the data if necessary, and transfers data words to the main memory buffer location designated by the address field of the descriptor. Then, under control of a main memory buffer-to-disc file descriptor, the input-output data service means delivers to the disc file controller a disc file address and main memory buffer address and indicates whether an input or output transfer is to occur. The data service means then transfers the data from main memory buffer area to disc file. The sequence is repeated until all the incoming information is accumulated in the disc file. Upon completion of the disc file transfers, which may be ascertained by counting the extent of the accumulated information or by examining the incoming data, or may be indicated by the programs for this channel of the data service computer, or by a signal received from the input peripheral device, all relevant channels return to their program and parameter areas in main memory for further tasks which may have been placed there by the executive program. In the case of the present example, the disc channel may find instructions to transfer the assembled data to a specific main memory address. The transfer of information in the reverse direction to an output

device, either from disc file, or from main memory via disc file, follows a similar procedure but in reverse order.

A further feature of the input-output data service computer is its capability to successively access input-output jobs stored in a main memory to be performed by the input-output channels after the channels have completed previously assigned jobs. Upon obtaining a job from a job stack stored in main memory, the input-output processor unit acquires a new state word bearing parameter and program instruction addressing information and stores the same in the local memory unit. This access of and response to input-output jobs stored in an external memory is completely independent of central processor interference and is completely under control of the input-output data service computer, requiring no interference with a central processor. The input-output jobs may refer to transfers between input or output devices and main memory or may direct data transfers between input and output devices themselves, each of which may include intermediate transfers which themselves may comprise several steps in the information transfer specified.

In order that the disclosure will be more fully understood and readily carried into effect, the following detailed description is given with reference to the accompanying drawings in which:

Fig. 1 is a block diagram of the input-output data service computer of the subject invention;

Fig. 2 is a block diagram of the I/O processing unit of the input-output data service computer of the invention;

Fig. 3 is a format drawing showing the manner in which the drawings of Figs. 3A to 3F are to be arranged to form a detailed schematic block diagram of the invention, Figs. 3A to 3C being the schematic block diagram of the I/O processing unit, Figs. 3D and 3E being the block diagrams of the local memory unit and the data service unit, and Fig. 3F being the detailed schematic block diagram of the communication unit of the invention;

Fig. 4 is a detailed block diagram of the contemplated physical arrangement of several of the processing unit registers which communicate directly with the operand registers of the arithmetic unit and the address modifying section of the input-output processing unit;

Fig. 5A is a detailed block diagram of the local memory unit of the invention, Fig. 5B illustrates word formats of the channel controls, and Fig. 5C illustrates alternative apparatus for monitoring and servicing program flags in the I/O processing unit;

Fig. 6 shows the logic gates of the byte length and position decoder of the invention;

Fig. 7, which consists of Figs. 7A through 7H and 7J through 7M, and Fig. 8 when read together illustrate the byte insertion or selection gating of the invention;

Fig. 9, which consists of Figs. 9A through 9F, shows alternative byte insertion or selection gating for the subject invention;

Figs. 10 and 11 when read together constitute the priority resolver and address encoder of the subject invention; Fig. 10 consisting of Figs. 10A through 10F and 10AA through 10AC, Fig. 10A being a format drawing showing the manner in which Figs. 10AA to 10AC are to be arranged, and Fig. 11 being a format drawing showing the manner in which Figs. 11A to 11C are to be arranged;

Fig. 12 shows a modular data processing system in which the input-output data service computer of this invention is shown connected as a module; and

Fig. 13 shows the interconnections provided between the input-output data service computer and other components when utilized in the modular data processing system of Fig. 12.

In Fig. 1 there is shown a block diagram of the input-output data service computer of the subject invention. Input-output processing unit 20 and data service unit 40 are each connected to local memory unit 30 by way of interconnections designated 29. Either the processing unit or the data service unit may make access to the local memory unit for writing or reading word portions of the channel controls so long as the memory unit is not busy servicing the other. Processing unit 20 and data service unit 40 may also each communicate through interconnections 31 with external memory communications unit 50, as shown. The communications unit contains a memory conflict resolver for awarding priority of communication with main memory modules 55 to the data service unit in order to avoid loss of data when both the processing unit and the data service unit are seeking access.

As shown, the I/O processing unit also has connections with central processor modules 25 when used in a system. Certain interrupt signals are communicated to the central processor over cable 24 and a main memory job stack address is communicated from the central processor to the I/O processing unit over cables 26. The processing unit includes a job stack address register for storing this main memory address from which new jobs can be obtained. Once a job has been received and a state word containing addressing information is stored into the local memory unit of the processing unit, the processing unit is then able to access program routines stored in an external memory by local memory to main

memory relative addressing through the communications unit. Access to program instructions by the processing unit occurs upon the detection of program flag signals and the grant of priority of program service to one of the input-output channels. The processing unit executes the programmed instructions, utilizing its adders and comparators under control of operand field discrimination apparatus for constructing a channel descriptor as directed by the program, and stores the descriptor so constructed in the local memory unit.

Local memory unit 30 contains memory locations for storing a channel control for each of the input-output channels. Each channel control utilizes two addresses in the memory unit: one for the storage of a state word containing main memory addressing information, and the other for the storage of a descriptor which describes fully an information transfer to be performed, in addition to space for the buffering of input or output data bytes for the packing or unpacking of full data words for communication with main memory. Also shown included in the local memory unit in Figure 3D are a memory write register, a memory information register, a memory address register and address decoder.

Data service unit 40 is connected to peripheral equipment controllers 45 which include I/O buffer submodules. The connections to the controllers of simplex output devices such as printers, card and tape punches and display devices are the cables designated 44. Normal communication between the central processor and the main memory have been omitted from Figure 1 for reasons of clarity. The connections to the controllers of simplex input devices such as punched card and punched paper tape readers and character recognition devices are the cables designated 46. Connections between the data service unit and the controllers for half duplex devices, such as magnetic tape units, disc files, or drums, or combined paper tape punching and reading apparatus, or teletype-writers could be effected over either cables 44 or 46, provided the data lines are able to pass data in either direction, one direction at a time. Full duplex input-output devices such as some teletype units, and some types of tape unit controllers, would utilize both interconnection cables 44 and 46 simultaneously.

Data service unit 40 monitors and resolves priority of input-output channel service requests, decoding and implementing channel descriptors directly from the local memory information register. Information transfer paths are provided for enabling information transfer between input and output devices and the local memory data buffer area or a main memory module through the communications unit. Also incorporated in the data service unit are a service request look-ahead capability for advanced strobing of input device controllers, device start line controls, and counters and controls for enabling and regulating input and output transfers to and from the device controllers.

Communications unit 50, in addition to the memory conflict resolver previously discussed, contains word storage registers, communication memory address registers, store and fetch registers, communication word counters, timing and controls. All data transmitted to and from the main memory as well as the fetching of new state words and program instructions and the storing of interrupt signal information is effected by communications unit 50 to main memory 55 over interconnection cabling 54, as shown. This information transfer may be a one word transfer or a sequence of words.

Fig. 2 is a block diagram of the I/O processing unit of the novel input-output data service computer. As illustrated, the processing unit consists of a program flag responder 21, an arithmetic unit 22, and a control unit 23, which includes a subcommand matrix. The program flag responder contains a program flag flip-flop for each input and output channel and a flip-flop for the job stack program flag from the central processor which signals the presence of new jobs in an externally stored job stack, the address of which the I/O processor control unit stores in its job strap address register. Also included in the program flag responder is a priority resolver and address encoder for servicing the channel program flags and a channel identity encoder for resetting the program flags once serviced and for setting the input-output channel start line flip-flops, which may be reset under program control. As shown, the program flag responder communicates channel identity address information to the control unit which it stores in the channel base address register and receives subcommands from the control unit including commands to reset the program flags as they are serviced.

Control unit 23 contains a job stack address register for storing the location of externally stored input-output jobs which are to be successively accessed and performed by the data service computer and a state word register for storing new state words provided by new job stack entries from main memory. The control unit also contains an address modifier and effective address register for use in addressing both the local memory unit and the main memory unit. Upon obtaining instructions from main memory through the communications unit, storing them in the func-

tion register extension, and unpacking the instruction syllables and placing them in the function register, the control unit executes the instructions and addresses parameters previously placed in main memory for use in executing the assigned program.

The instructions of the program designated for the channel are executed by arithmetic unit 22 which includes addressable operand register AOR and buffer operand register BOR operating in conjunction with temporary operand storage registers, adders, and a comparator as indicated. Addressable operand register AOR also serves as an accumulator since the result of adding operands from the AOR and the BOR in the adders is returned to the AOR. The arithmetic unit also contains a field discriminator connected to the two operand registers for defining the field of the operands to be added or compared as directed by the instructions being executed.

The primary function of the arithmetic unit is to construct descriptors for describing information transfers under control of subcommands received from the control unit which are developed by decoding instructions obtained from a program routine in main memory. The arithmetic unit can access parameters directly from an external main memory through the communications unit and may transfer operands and results back through the communications unit. Operands can also be obtained from the local memory and either returned to local memory or modified to construct a new descriptor to be placed in the appropriate channel control in local memory.

A detailed schematic block diagram of the invention is illustrated in the drawings of Figs. 3A through 3F which are to be read as one drawing as illustrated in the format drawing of Fig. 3. Figs. 3A through 3C contain a schematic block diagram of the I/O processing unit. Figs. 3D and 3E contain the block diagrams of the local memory unit and the data service unit of the invention, and Fig. 3F is a detailed schematic block diagram of the communication unit of the input-output data service computer.

The I/O processor unit as shown in Fig. 3 contains a program flag flip-flop for each input-output channel and a job stack flag flip-flop, designated 101, coupled to priority resolver and address encoder 102 which is coupled to deliver a channel address to channel base address register 103. Channel decoder 105 ascertains channel identity for setting the appropriate start line flip-flop 171 once the I/O processor has begun servicing a channel and resets the program flag 101 which was recognized. The channel decoder can also set any program flip-flop 101 and reset any start line flip-flop 171 under program control.

CHBR 103 is also connected to address modification base selector 112 for providing a local memory channel address or developing such an address in address modifying adder 114 by addition with the constant 001 or 010 supplied through address modifier selector 111. The channel control address thereby obtained is delivered to effective address register 116 for addressing the selected portion of a channel control, i.e., the state word, index register, or descriptor. The address from the effective address register is transmitted to local memory address register 166 and is decoded in address decoder 167 for causing the appropriate channel control portion to be brought into memory information register 162 and subsequently delivered to buffer operand register 120 from which it can be transmitted to addressable operand register 122.

Along with buffer operand register (BOR) 120 and addressable operand register (AOR) 122, the arithmetic unit of the I/O processor contains addressable storage register (HR) 123, temporary storage register (CR) 124, adders 126, comparator 127, and field discriminator 129. All communications into the arithmetic unit from the local memory unit and the external main memory through the communications unit are received by the buffer operand register BOR. All communications from the arithmetic unit to the local memory unit and to an external memory through the communications unit are by way of the addressable operand register AOR or the addressable storage register HR.

Once a state word is addressed in the local memory unit by the I/O processor unit, it is delivered by way of the BOR and the AOR to state word register 130. This register contains a portion for storing a parameter base address 131, an instruction base address 133, an instruction counter 134, and an activity bit. Associated with the state word register are index registers X1R 136 and X2R 137. Instructions in main memory are addressed by summing the instruction base address through address modification base selector 112 and the instruction counter through address modifier selector 111 in address modifying adder 114. The instruction address is transmitted to effective address register 116 and subsequently to communications address register 204 for delivery to main memory through communication unit drivers 212. The memory operation code for addressing instructions is loaded into the memory operation code area of communication address register 204 by subcommand matrix 150 of the I/O processing unit.

The instructions obtained from main

memory may, for example, direct the I/O processor unit to construct a new descriptor for the input-output channel being serviced, may direct the processor to modify a channel state word, or may contain a channel control command such as "stop channel" when a complete input-output transfer job has been performed for resetting the active bit in the state word register, may order the resetting of a start line flip-flop upon detection of an error or malfunction in the operation of the device, or may command that the immediately preceding channel descriptor for a device be reinstated for repeating a transfer which was unsuccessfully attempted. Thus, upon detection of a parity error in writing upon or reading from a magnetic tape unit, for example, a program subroutine may order that the tape be rewound and that the write or read operation be updated. This repetition of incompletely performed information transfers may be repeated as many times in the program as desired, after which an instruction will direct the I/O subcommand matrix of the processing unit to send an interrupt information word for storage in the location in main memory identified by interrupt stack address register 157. This interrupt word, which is sent to main memory through the communication unit, includes the number of the input-output channel involved and the reason for termination, which could be, for example, input-output information transfer completed successfully, input-output device malfunction, printer buffer overflow, parity error detection from device, or an indication that more disc file or main memory locations are needed for completion of an information transfer. If a parity error is detected by the input-output communication unit in connection with reading from main memory, the I/O processor unit will send a parity error interrupt signal to the system central processor.

If the instruction received from main memory denotes completion of a transfer job by stopping the channel and resetting the active bit, the I/O processor subcommand matrix will send an I O job completed interrupt to the central processor. This interrupt line signals that an interrupt stack entry has been sent by the I O processing unit through the communication unit to the interrupt stack address location in main memory identified by the address in interrupt stack address register 157. The data processing system scheduling program is thereby signalled that a transfer job is completed on a particular input-output channel and that the channel can be assigned a new job by the central processor. When a new job is assigned, the central processor raises a job stack flag flip-flop in program flag flip-flops 101, which upon being granted service by the I/O processor unit will provide directly or indirectly a new state word for the channel to be operated for performing the job.

The instructions obtained from main memory through communication unit receivers 215 and communication latch register 217 are delivered to function register extension 140 by way of buffer operand register 120. Function register extension FRE 140 is a four word buffer register for receiving sequentially four instruction-containing words and providing syllables as needed to function register 145 under control of unpacking and formating device 142 and syllable counters and controls 143. As shown in the drawing, the function register extension FRE is capable of receiving and storing instruction-containing words having eight syllables. Since instructions require only one to six syllables as illustrated by function register 145, each instruction word obtained from main memory and stored in FRE 140 may contain parts of two or more instructions which will be unpacked and formated into function register 145 for controlling the operation of the I/O processing unit.

All instructions in the instruction repertoire of the input-output data service computer will contain as the first syllable an operation code which identifies the operation to be executed, such as load, store, add, complement, subtract, shift, jump, skip, transfer, subroutine jump or return, go, delay, link, or release processor. Also, in the preferred embodiment, a portion of the fifth syllable and the entire sixth syllable contains an address modifier to be used by the processor in obtaining parameters or operands for use in executing the instruction. The remaining portion of the fifth syllable contains a code for referencing the instruction to one or the other of index registers 136 and 137. Syllable four contains an address modifier P which may be used by the instruction for modifying the address appearing in syllables five and six. The second syllable of the instructions may contain shift control signals for directing that addressable operand register AOR be shifted by shift matrix 151 and to what extent in response to signals generated by shift control counters 147 and 148 which are decoded by decoder 149. In the preferred embodiment SCCA is given the number of one-place shifts that are to be performed in AOR 122 and shift control counter B 148 is given the number of octal shifts to be made in the AOR. Thus, for an addressable operand register containing 52 bits, seven clock pulses at most are necessary for shifting this register from one to fifty-one positions within the register.

The second and third syllables of the

instruction format can alternatively contain first field bit and last field bit information for delivery to field discriminator 129 for controlling the fields of operand registers 120 and 122 which are to be utilized by the arithmetic unit. The instruction can designate that an operation be performed utilizing the field included between the first and last bit positions and may retain intact the excluded field. The second and third instruction syllables also may contain signals designating a shift or jump or a special variant syllable which may specify the direction of transfer between registers of the I/O processing unit or may contain a control signal for increasing or decreasing one or the other of index registers 136 and 137. This special variant syllable may be inverted in inverter 146 and delivered to address modifier selector 111 for modifying either index registers 136 or 137. The contents of the index register to be modified are delivered to address modification base selector 112 for being modified in address modifying adder 114 and returned to the index register by way of effective address register 116.

The most frequent use of the I/O processing unit is in the construction of descriptors in response to a set of instructions from a designated program in main memory which is assigned to the channel by the system executive scheduling program. It is important to note that the invention permits the assigment of more than one input-output channel to each of the programs and program routines stored in main memory since the I/O processing unit services only one channel at a time and since instructions and parameters are separately stored and separately addressed. Thus, many similar devices may be assigned to one program which will be identified in the instruction base address field of their state words since each may be assigned to a unique set of parameters by the parameter base address field of each channel word. The specific transfer to be accomplished by the channel, the amount of information to be transferred and the address to which the information is to be transferred, etc., can be assigned by the creation of a table of parameters in the main memory, there being a unique table for each input-output channel. The common use of program routines for the servicing of more than one of the input-output channels reduces the memory capacity required for storing programs in main memory. The multiple use of program routines also reduces to a minimum the number of programs which must be set up and permits the scheduling program to simply utilize available program routines which have been previously debugged, tested and simplified, rather than having to rewrite or recopy the program routines when assigning new information transfer jobs to the input-output channels.

In the preferred embodiment it has been found to be advantageous to group together the temporary storage registers, the entire state word register and associated index registers 136 and 137, and interrupt stack register 158 in order to minimize signal transfer paths and to reduce the space required for these registers in the processing unit. In the embodiment shown in Fig. 4 which utilizes 48 bit temporary storage registers and 18 bit index registers, and instruction and parameter base addresses, ten bit instruction counters, and a ten bit interrupt stack limit register, it has been found that separate grouping of these registers into subgroups 220, 230, 240 and 250 is desirable. As can be seen in the figure, each of the subgroups has inputs from addressable operand register AOR and buffer operand register BOR, in addition to inputs from the effective address register, the address modification sum, or a word counter in some instances.

Subgroup 220 is a two bit portion of the register grouping which contains the first two bits of temporary registers C and H plus the one bit activity indication 139, as illustrated. All other positions are unused or not applicable. The inputs are an AOR bit signal or a BOR bit signal which are gated as indicated by command signals in AND gates 221 and 222. Signal input line 227 is a representative signal line which corresponds to the second bit in each of the registers. Another input signal line would be provided for the first bit position. An output line is provided from the second bit position to the corresponding AOR bit position as indicated. Selection of the register to be effected by the input is accomplished under control of the write control signals W or the enable read control signals E provided to each register portion of each subgroup.

Subgroup 230 comprises ten bit positions of C register 124 and H register 123, as well as ISLR 158 and instruction counter 134, including its last two bit positions which are designated 135. The input to this subgroup over representative input signal line 237 is an ORed function of gated signals from the AOR, BOR, address modifier sum and in the case of the least two significant bits of the instruction counter, a word counter. The output of this subgroup is transmitted to appropriate bit positions in the AOR, and in the last two positions, to the word counter. After being gated with enable read control signals in AND gate 238, the output is delivered to an appropriate position in address modifier selector 111 which also receives

gated inputs from a gated function of EAR + ISA + JSA + $\overline{FR(V)}$ + FR(INC) as illustrated.

The third subgroup 240 contains 18 bit positions of CR and HR in addition to index register 137 and the instruction base address register 133. Gated inputs from the AOR, the BOR, and the EAR are ORed to representative input line 247. An output is provided from each position to a comparable AOR position and is also delivered to an address modification base selector position after being gated with enable read signals in AND gate 246.

Subgroup 250 contains 18 bit positions of the CR and HR in addition to index register X1R 136 and the parameter base address 131. Inputs from the AOR, EAR, AM sum, and the BOR are gated by control command signals and then ORed together over representative input line 257. Each position provides an output to a corresponding AOR position and after being gated with enable read control signals 256 are outputted to an address modification base selector position, as is also a gated signal from CHBR 103 and a corresponding position in the AOR as indicated.

The primary function of the I/O processing unit is to construct descriptors each of which describes a specific information transfer to be performed in response to instructions provided by an external addressable program store and thereafter transmit the descriptor so constructed to the local memory unit. The descriptor is delivered to local memory write register 164 from the AOR and written into local memory thin film stack 160 at the address location transmitted to local memory address register 166 by effective address register 116. The transferring and storing of a descriptor by the processing unit will occur only if the memory is not busy under control of the data service unit for servicing a data transfer. After delivery of a newly constructed descriptor to the local memory by the I/O processor, the next program instruction may release the processor from the service of that program and enable it to scan program flag flip-flops 101 for servicing programs on other channels.

The I/O processing unit shares the local memory unit with the data service unit but otherwise operates independently of and asychronously to the data service unit. The I/O processor also shares the communication unit with the data service unit, access priority being controlled by the operation of timing and control 200 and indicators 201 and 202 in conjunction with communication priority resolver 203.

A detailed representation of the manner of addressing a channel control in the local memory unit is illustrated in Fig. 5A. As shown, thin film memory stack 260 contains for each channel control a state word, an index register field, a descriptor, and a data field. The state word and indexes are stored as one memory word in the local memory unit. The descripter and data field are also stored as one memory word. Memory write register 264 and memory information register 262 are both capable of transmitting separately any of the four portions of a channel control.

In the preferred embodiment there are 512 input-output channels and therefore nine bits are required for addressing any one of the channel controls. The address stored in the channel base register therefore contains nine bits which are inserted into the 8 through 16 bit positions in address modifying adder 114 and effective address register 116. In order to select a specific portion of a channel control, a constant is inserted into address modifying adder 114 at bit positions 17 and 18 and transmitted to effective address register 116.

Ten of the address location bits are then transmitted by the EAR directly to local memory address register 266 and to address decoder 267 for selecting one word of a channel control consisting of a state word and index field or a descriptor and data field. The least significant bit of the local memory address, which is bit 18 in the effective address register, is presented, along with its inverse from inverter 118, to control gates 259, 261 for reading out the desired portion of the selected channel control to BOR 120 or to gates 263 and 265 respectively for writing one or both portions of the selected channel control in the memory unit from A operand register 122. Provision is also made for copying a portion of a channel control back into the local memory unit through gates 268 and 270 in response to copy memory signals CMB or CMA upon being gated by the control signals CB or CA.

Fig. 5B is a detailed illustration of sample channel control 161, there being 512 channel controls in the preferred embodiment. Each channel control contains a state word control designated 161-1 and a descriptor word control designated 161-2.

State word control 161-1 contains a state word, index registers and an unused word portion. The state word itself comprises an instruction counter IC, an instruction base address IBA, and a parameter base address PBA. Index registers X1R and X2R and an unused portion constitute the remainder of the state word control 161-1X as shown.

As illustrated, descriptor word control 161-2 contains a descriptor word and a data buffer field. A descriptor comprises

the following information: byte size code, byte position code, control code, byte count, device status code, and main memory address. The remainder of the descriptor word 161-2X is available for use as a data byte buffer by the data service unit.

Fig. 5C illustrates an alternative apparatus for monitoring and servicing program flags in the I-O processing unit which may be substituted in place of priority resolver and address encoder 102 and program flag flip-flops 101. In this embodiment the program flags, in the form of the numbers of the channels which are awaiting program service by the processing unit, are stacked in the unused portion of the state word controls in the local memory unit in the order in which they arise. Two channel address registers, channel address counter A 106 and channel address counter B 107, are utilized for recording the existence and location of program flags in the memory unit and are used in addressing the program flags.

Channel address counter A 106 stores the local memory unit address of the next program flag to be serviced and channel address counter B 107 stores the next vacant program flag address in the state word control portion of the local memory unit. The addresses in counters A and B are tested for equality in comparator 108, and if unequal, the channel address located in counter A 106 is placed into channel base address register CHBR and counter A 106 is upcounted by one at the same time.

In the initial state, channel address counters A 106 and B 107 are both zero. The processor begins operation upon recognizing a job stack flag from the central processor in job flag flip-flop 109 and fetches a state word from main memory and may up-count channel address counter B 107 by one. Comparator 108 then detects inequality between channel address counters A and B and the processor subcommand matrix delivers the channel address in counter A 106 into channel base address register CHBR and at the same time up-counts channel address counter A 106. The processor will proceed to fetch the channel number stored in the zero channel position in the local memory unit and enter it into CHBR since channel address counter A had read zero. The processor will then fetch the state word associated with the channel number received and will execute the program of that channel.

Channel address counter B 107 can be up-counted either by the processor unit under program control through channel decoder 105, or by the data service unit upon channel descripter exhaustion, as indicated. Every time the channel address counters are unequal as determined by comparator 108, the processor unit will go to the portion of the state word control designated for program flags for obtaining the number of the channel which is to be next granted program execution service. Thus the state word control portion of the local memory unit which is designated "UNUSED" in Fig. 5B will be operated as a first in—first out (FIFO) program flag storage stack of the identification of channels to be given program service. Each time a program flag address is taken from channel address counter A 106, that counter is up-counted by one so that it will point to the next program flag which will be the number of the device to be next given program service.

An important feature of this embodiment is a provision for giving priority to certain high signal rate input-output devices such as disc files which transfer data in groups or bytes of eight words each. This priority is granted to the disc files by down-counting channel address counter A 106 upon recognition that a program service is required for such a channel and the channel number of that device is inserted into the address in the local memory unit of the down-counted contents of the channel address counter A 106. Therefore, the processor unit upon detecting inequality between counters A and B, will first address the program flag of the disc file which is located at the address indicated by counter A regardless of whether other program flags are stored in other addresses in the local memory unit.

Returning to Fig. 3, service of the descriptors is placed in local memory stack 160 and performed by the data service unit. The first prerequisite to servicing an input-output channel is that the start line flip-flop 171 for the input-output channel must have been set by the program. This flip-flop is set from channel decoder 105 in the processing unit, as shown.

The device to be operated, whether it be an inputting device or an outputting device, raises its service request flag when available for the transfer of a byte of information and sets its service request flip-flop 172. Priority resolver and address encoder 173 scans the service request flip-flops and, upon upwarding priority and encoding an address, sends a signal which starts counters and controls 174 of the data service unit as illustrated and transmits the address of the channel given recognition to channel look-ahead register 175. The channel address is examined by decoder 176 and if designating an input device and if the input byte is present in the device controller, an input select strobe is sent to the device controller to strobe the byte into input register 193. Decoder 177 at this time also determines if a four or eight word inputting device is involved, and if so, it enables gate 178 from permitting the inputted informa-

tion to pass directly from receivers 192 through input register 193 to the communication unit, thus bypassing byte insertion or selection gating 191.

As soon as the local memory unit is not busy, indicator 169 enables gate 179 and permits the recognized input-output channel address to be transmitted to processing address register 180.

A signal is sent to the address register of the local memory unit for accessing the descriptor word, including the data buffer portion, into memory information register 162. The byte size and byte position codes are delivered to byte length and position decoder 190 for enabling gating 191 to insert the byte into local memory write register 164 for inputting devices or to select a byte from memory information register 162 for delivery to an outputting device through output register 196 and drivers 197. Decoder 181 examines the channel address in PA register 180 and delivers an output select strobe to the device controller if it serves an output device which accepts bytes less than or equal to a word of data. Decoder 182 sends signals to counters and controls 174 which indicate the amount of data to be sent out and provides an enabling signal to byte length and position decoder 190 if the transfer is to be a word or less. A signal is generated at this time by counters and controls 174 for strobing the data byte from the input register to local memory write register 164 or to communication storage register 206 or 207 or from local memory information register 162 or communication fetch register 217 to the output register. Control signals are generated by decoder 181 for resetting the start line flip-flop involved upon error, malfunction, or termination status detection and for setting the program flag flip-flop 101 for the channel upon exhaustion of the byte count field of the descriptor which signifies that the transfer of a block of information has been completed.

Upon receving a not-busy signal from either indicator 201 or 202 of the communication unit, the selected channel address is then delivered to output address register 185 by gate 183. The output address register is then decoded for delivering output select strobes to controllers of devices capable of receiving four or eight word outputs by decoder 186 upon the receipt of information from main memory through the communication fetch register to output register 196. Decoder 187 produces signals to indicate if a four or a eight word output transfer is involved, in which case gate 188 will be enabled to permit the direct transfer of information from communication unit fetch register 217 directly to drivers 197 through parity generator 195 and output register 196, thus bypassing byte insertion or selection gating 191.

Parity check 194 examines parity of information received from inputting information devices and delivers its signal to counters and controls 174. A status signal from the output device controllers is delivered to counters and controls 174 to enable counters and controls 174 to prevent information transfer to inoperative or disconnect devices.

The communications unit has two storage registers 208 and 209 associated with address registers 206 and 207. Indicators 201 and 202 indicate whether information is accumulated in the storage registers awaiting delivery to a main memory module. Both information and memory address signals are communicated to the main memory module by way of communication storage register 211 and drivers 212. The memory address is sent from either effective address register 166 of the processing unit or from memory information register 162 by way of X and Y address registers 206 and 207 to communication address register 204 and then to communication store register 211 through parity generator 210. In addressing a main memory module, the memory module code portion of communication address register 204 is examined by decoder 214 for sending a request to the memory module involved. Upon receiving an acknowledging signal from the memory module through receivers 215 into timing and control 200, the memory module address is transmitted to the memory module by drivers 212, followed by the information signals from the communication storage register. Decoder 213 examines the memory operation code in communication address register 204 to determine the number of words to be included in the transfer. The parity of information signals received from main memory by receivers 215 and communication fetch register 217 is examined by parity check 218 which signals timing and control 200 directly.

Upon transfer of a byte of information between the local memory unit and a device controller or between the main memory unit and a device controller, descriptor field modifying counter 199 updates the descriptor by taking fields from memory information register 162, modifying them, and returning them to local memory write register 164. Upon updating the channel descriptor and returning it to the local memory thin film stack, the local memory unit is released and the data service unit is available to service the next input-output device having priority in accordance with its descriptor as soon as the local memory unit is not busy servicing the I-O processing unit.

Interrupt stack limit register ISLR 158, which records the number of entries avail-

able in the main memory interrupt stack, is decreased by one each time an interrupt entry is made. Upon the detection that the ISL is zero by decoder 159, no more entries are attempted in the stack. Decoder 117 signals the I/O subcommand matrix when the contents of effective address register 116 equal zero. This decoder may be used for testing the index registers when desired under program control.

Time counter 153 and phase counter 154 provide control signals to the subcommand matrix for clocking the cycles of operation of the input-output processor. Control flip-flops 156, which may be set and reset under program control as indicated, control the operation of the processing unit in skipping or jumping in program instructions as in the subroutine jump and the skip instructions. The setting of the mode flip-flop specifies whether the I/O processor is engaged in executing channel programs or is accessing new state words from the external job stack.

Fig. 6 shows the logic gates of the byte length and position decoder of the invention which is designated 190 in Fig. 3D. The inputs to gates 301 are the first three bit positions of the channel descriptor which comprises the bit length code and the inputs to gates 302 are the next three bit positions of the descriptor which constitutes the bit position code. The inputs to these decoder gates are taken from memory information register 162 as shown in Fig. 3D. Six of the eight possible states of the bit length code are used in gates 301 to produce a signal for each size of byte length to be utilized in the input-output data service computer. All eight different permutations of the bits of the bit position code are utilized by gates 302-0 through 302-7 to produce signals for use in specifying the position for insertion or selection of a byte of one of the specified byte lengths.

Fig. 7, which consists of Figs. 7A through 7M omitting 7I, and Fig. 8, when read together, illustrate the byte insertion or selection gating of the invention which is designated 191 in Fig. 3D. Each of Figs. 7A through 7H and 7J through 7M have six gates A through F for individually receiving the six bit length signals produced by decoder gates 301 of Fig. 6. In Fig. 7A, gates 311A through 311E each receive the bit position signal BP0 as an input, gate 311F having only BL 48 as its input. The outputs of the AND gates 311 are ORed together to produce a signal which indicates that a byte is to be placed or taken from the first six bit positions of the data buffer word BLP1-6. Inverter 310-1 is utilized for providing the inverse of this bit length and position signal.

In Fig. 7B, BP1 is an input of gate 312A and BP0 is an input of gates 312B through E, while gate 312F has only BL 48 as an input. The ORed output of gates 312 is BLP7-8 which is inverted in inverter 310-2 for supplying the inverse of the signal. BP1 and BP0 are also used as inputs to gates 313 of Fig. 7C as shown. The output of gates 313 are ORed to produce the BLP 9-12 signal and inverted to 310-3 to produce the inverse signal.

The bit position signals BP2, BP1 and BP0 are used as inputs to gates 314 of Fig. 7D and 315 of Fig. 7E. The outputs of these AND gates are ORed together to produce the BLP 13-16 signal and through inverter 310-4 its inverse in Fig. 7D and to produce BLP 17-18 and its inverse by inverter 310-5 in Fig. 7E.

The bit position signals to Fig. 7F AND gates 316 are BP3, BP2, BP1 and BP0. The outputs produced are BLP 19-24 and its inverse. In Fig. 7G, bit position signals BP4, BP3, BP2 and BP1 are used as inputs to AND gates 317 to produce output signals BLP 25-30 and its inverse. Bit position signals BP5, BP3, BP2, and BP1 are utilized as inputs to AND gates 318 in Fig. 7H to produce output signals BLP 31-32 and its inverse through inverter 310-8.

Fig. 7J utilizes BP5, BP4, BP2, and BP1 as inputs to AND gates 319 for producing BLP 33-36 and its inverse. AND gates 320 of Fig. 7K utilize BP6, BP4, BP3, BP2, BP1 as inputs to produce output signal BLP 37-40 and its inverse. Bit position signals BP6, BP5, BP3, BP2, BP1 are utilized as inputs to AND gates 321 of Fig. 7L to produce output signal BLP 41-42 and its inverse. AND gates 322 of Fig. 7M utilize as inputs BP7, BP5, BP3, BP2 and BP1 to produce output signal BLP 43-48 and its inverse through inverter 310-12.

Fig. 8 consisting of Fig. 8A and 8B, shows the gates which enable selective insertion or selection of bit positions within a desired byte. In Fig. 8A a gate 331 receives inputs from a corresponding bit signal from memory information register 162 and one of the bit length and position signals developed in Figs. 7A through 7M, for sending a bit from the memory information register to the output register upon actuation of a control. Gate 332 receives one of the bit length and position signals from Figs. 7A through 7M and a corresponding bit signal from communication fetch register 217 for sending a bit position from that register to the output register upon receipt of a control signal. AND gates 331 and 332 of Fig. 8A are the final gates of the output selection gating apparatus.

In Fig. 8B, a gate 333 receives one of the bit length position signals and a bit signal from the input register for sending an information bit to the local memory stack

or main memory buffer. Gate 334 receives the inverse of the bit length and position signal and a bit signal from the memory information register for sending a bit from that memory register to the local memory stack or main memory buffer. Gates 333 and 334 are the final gates of the input insertion gating apparatus.

Figs. 9A through 9F show alternative bit insertion or selection gating designated 191 in Fig. 3D. By receiving input signals from bit length gates 301 and bit position gates 302 of Fig. 6, the AND gates 341 of Fig. 9A produce bit length and position signals BLP 1-6 and BLP 7-8, buffers 340-1 and 340-2 being used to equalize signal magnitudes as shown. The inverse of these bit length and position signals can be obtained by use of inverters as in Figs. 7A through 7M. Gates 342 of Fig. 9B gate together bit position signals and bit length signals to produce BLP 9-12 and BLP 13-16. These may be inverted as needed. AND gates 343 of Fig. 9C receive bit length and bit position signals to produce BLP 17-18 and BLP 19-24. AND gates 344 of Fig. 9D receive bit length and bit position signals to produce BLP 25-30 and BLP 31-32. AND gates 345 and 356 of Figs. 9E and 9F receive bit length and bit position signals to produce respectively BLP 33-36, BLP 37-40, and BLP 41-42, BLP 43-48. All buffers 340 utilized in Figs. 9A through 9F are similar and are used to equalize signals and loading in the circuits.

The bit length and position signals produced by the gating circuits of Figs. 9A through 9F may be used as the BLP inputs to the gates of Figs. 8A and 8B, which constitute the final gates of the input insertion and the output selection gates of the invention. The inverted bit length and position signal required as an input to gate 334 of Fig. 8B can be obtained by inverting the appropriate BLP signal of the circuits of Figs. 9A through 9F.

Figs. 10 and 11 when read together constitute the priority resolver and address encoder of the subject invention which is designated 173 in the data service unit as shown in Fig. 3D and designated 102 in the input-output processing unit as shown in Fig. 3B. Fig. 10 consists of Figs. 10A through 10F and 10AA through 10AC. Fig. 10A being a format drawing showing the manner in which Figs. 10AA to 10AC are to be arranged to form the detailed logic gating for the gates of Fig. 11.

In Fig. 10A, which consists of Figs. 10AA through 10AC, the first 64 flag flip-flops and their inverse signals are gated to produce a signal designated A91, which will be high for any flip-flop whose binary equivalent number ends in a "1", that is, a flag given an odd decimal number. Buffer elements 351, 352 and 353 are utilized for buffering signals obtained from gating the first 8 flags, the next 24 flags, and the last 32 flags which signals are ORed together to form the A91 signal. There will be a separate A9 signal generated for each succeeding block of 64 flags, there being a total of eight blocks in all if 512 flags are to be serviced.

Fig. 10B shows a gating circuit comprising AND gates 360A through 360R for generating a signal A81 which is the ORed result of the outputs of buffers 361 and 362. A81 will be high for every flag having a "1" in the next to the least significant bit position of the equivalent binary number. Fig. 10C shows a gating circuit for developing a signal A71 through buffer element 371 utilizing gates 370A through H. A71 will be high for any flag whose binary equivalent number has a "1" in the third least significant bit position. Fig. 10D shows a gating circuit utilizing gates 380A through D for developing a signal A61 through buffer element 381. A61 will be high for any flags having a "1" in the fourth least significant bit position. Fig. 10E shows a gating circuit having gates 391 and 392 for producing a signal A51 which will be high for the first 16 and third 16 flip-flops of each group of 64 flags. Fig. 10F illustrates that a signal A41, produced through buffer element 396, will be high for the first 32 flip-flops in each group of 64. As in the development of A91 in Fig. 10A, there will be eight sets of gates similar to Figs. 10B through 10F for developing corresponding signals for each of the eight groups of 64 flags, which would be present in a total of 512 flags.

Fig. 11 which consists of Figs. 11A through C illustrates gating means for developing a signal corresponding to the particular group of 64 in which a flag appears. These group signals are ANDed with the corresponding A9, A8, A7, A6, A5, and A4 signals and then clocked to produced clocked A9, A8, A7, A6, A5, and A4 signals which are the six least significant bit positions of the binary address code for the flag to be recognized. The group signals are further selectively ORed together and clocked in gates 503, 502 and 501 to produce clocked A3, A2 and A1 signals which are construed as the three most significant bit positions of the address of the program flag given priority recognition by the priority resolver and address encoder.

Fig. 12 shows a modular data processing system in which the input-output data service computer of this invention is shown connected as an input-output control module. Such a system incorporates a central exchange to which are connected central processor modules CP1 to CP3 and

memory modules MM1 to MM16. Several of the input-output data service computers of the subject invention may be incorporated into the system, being identified in Fig. 12 as IOM1, IOM2 and IOM3. The input-output computer modules are shown connected to the peripheral device controllers by an input-output exchange which connects to, for example, disc file controllers, card reader controllers, magnetic tape controllers and teletype controllers.

Fig. 13 shows the interconnections provided between the input-output data service computer and other components when utilized in a modular data processing system as in Fig. 12. In Fig. 13 the input-output data service computer of the subject invention is identified as IOM and incorporates a number of sets of drivers and receivers for communicating with one or more central processor modules CPM, memory modules MM, a simplex input device controller SIDC, and a simplex output device controller SODC. The interconnecting cables transmit the necessary signals between the input-output module and other modes and peripheral device controllers in such a system. Cables having a circled number thereon illustrate a bus in each case which contains the circled number of connecting lines. It is important to note that for some of the modules, the input-output module contains a set of receivers for each one of that type of module, while the input-output module contains only one set of receivers for serving each of certain other modules of the same type through the cable bus.

It should be noted that a link instruction may be placed in a channel parameter area in main memory for directing the input-output processor to switch from processing a program on one I/O channel and begin executing that program on another channel in accordance with a set of parameters unique thereto. Execution of such link instructions enables this data service computer to perform a series of transfer jobs with similar devices without having to seek a new job state word for each transfer. Further, to be noted is that the I/O processor may be instructed to process information directly from main memory without processing a channel state word or directing any peripheral device transfers. This input-output computer may therefore, process data directly from main memory in parallel with the system central processor, if desired.

Description of the input-output data service computer operation:

The input-output data service computer relentlessly looks for work to be performed. Assuming that the I/O processor unit and the data service unit are both in their initial state, the I/O module waits for a signal from the central processor to start operations. Along with this signal and a job stack flag from the central processor, an 18 bit memory module address received by the I/O module is stored in the job stack address register JSAR, which points to the next state word in the job stack. The job stack in the main memory keeps a list of all the jobs to be executed by the I/O module in the form of state words. In response to the job stack flag the I/O processor fetches a state word from the job stack in the memory module and increments the job stack address register. The state word is stored in the state word register while the channel number is stored in the channel base address register. The instruction counter of the state word register is then set to zero so that the first program word will come from the address indicated in the instruction base address register.

The large repertoire of instructions for the I/O processing unit provides the I/O modules with decision making tools. Its prime function is to assemble descriptors using a memory module address to and from where data will be transferred. The descriptor also comprises a field which defines a byte count or number of consecutive bytes to be transferred, a field which defines the byte size and byte position, and a field which defines the terminating status, and finally some control bits. The memory module address defines the start of the buffer area in the memory module where the data is temporarily stored while going to or from the disc file memory. Two such buffered areas are used by each input or output device so that while the data transfer takes place between the first buffer and the disc file memory, a second data transfer takes place between the second buffer area and the slower input or output device.

Besides the data areas, the memory module also contains the program to be exercised by an input-output channel. The buffer areas are unique to each input-output device while the program can be shared by many input-output devices similar in nature. A parameter area is also unique to each input-output device, its base address being retained in the state word register. The parameter area keeps a list of the parameters which are unique to each device. This list contains the two memory module addresses for the buffer areas, the byte count which is inserted in the descriptor for controlling the transfer of each block of data, and perhaps the table which lists the actions that have taken place from the beginning of the job to the end.

The instruction repertoire of the I/O processor unit, unlike the general purpose computer, is oriented toward the functions that the I/O module most perform. The

I/O processor unit must operate on the many descriptor fields in order to form and check descriptors. The fields are defined in the instructions and the field discriminator allows addition, subtraction, comparison and other operations to be performed in the arithmetic unit.

Once the descriptor is formed, it is stored in the descriptor field in the channel controls in the local memory. Then the start line of the device whose number is contained in the channel base address register is raised. Finally the processor unit is released. The I/O processing unit is then free to determine whether a new entry exists in the job stack, and if so it will proceed to form a new descriptor from the new state word from the job stack until the list of jobs is exhausted.

Once the start line is raised, the device controller sends a service request to the data service of the I/O module informing the latter that it is ready to either send or receive a byte of data. Upon sensing the data service request from the device controller, the I/O data service unit reads the descriptor from the corresponding channel control in the local memory and allows the transfer of the byte. Then the descriptor is updated, that is, the byte count is decreased by one, and if the memory transfer took place, the memory address is also modified.

The I/O data service is designed to accept a variety of byte sizes. One byte may consist of a character of 6 bits or a character of 8 bits, or two, three, or four characters. These byte sizes are assigned to slow devices. For faster devices a byte size of one word or four words or eight words is assigned, the words consisting of 48 bits. When a byte size of less than one word is received in the data service unit, it is packed into the buffer word which is stored in local memory along with the descriptor. When this data word is full, it is then transferred to the buffer area in the memory module. For an output device which takes a byte transfer that is less than one word, then the reverse process occurs where the bytes are unpacked from the word which is fetched from the memory module and stored in the local memory temporarily.

The I/O data service unit is organized into many input and output busses. Since the data lines of many input devices are connected together, an input select strobe is sent to the input device which selects the data at the device controller and sends the information over the busses into the input register of the data service unit. This function is performed by the look-ahead address register and is controlled by the corresponding counters and controls. If the byte size is of one word or four words or eight words, it is guided directly to a storage register of the communication unit to be stored in an external memory module. The processing address register, called PA, has the function of reading the descriptor from the local memory, updating the memory address if necessary, and decrementing the byte count and updating the byte position again if necessary. In the case of an output device, the output byte is directed from the fetch register of the communication unit directly into the output register under the control of the output address register OA, and its counters and controls if the output byte was of one, four, or eight words. If the byte was less than one word, and if the data word in local memory was empty, then the word fetched at the memory address indicated in the descriptor through the communication unit is stored in local memory while the first byte is directed to the output register. This byte is defined by the byte position and byte length fields which are in the descriptor. In the case of an input byte of less than one word, the byte is inserted into the data word using the same byte length and position decoder.

Several service requests may be received by the data service unit simultaneously. However, priority will be resolved and a device which requires more frequent service will have priority over a device which requires less frequent service. This priority resolver has the function of encoding the requesting flip-flop level into a device number or channel address which is used to address the corresponding descriptor in local memory. Once a start line is raised in the controller of the device, the device then operates at its own speed in sending a service request to the I/O module when strobed and in buffering one byte of data while operating on the second byte. Between service requests of one device, the I/O data service unit responds to other service requests from other devices. The device controllers thus continue making data transfers until the descriptor is either exhausted or a terminating status has been detected. The descriptor is exhausted when the byte count reaches zero. The byte count of zero is a terminating status as well as a malfunction detected by either the device controller or the I/O data service unit. A terminating status may consist of a parity error detected in the transmission of data, a malfunction of the device itself, or a data too slow status to mention only a few of the terminating statuses.

When a terminating status is detected by the data service unit, it is then a function of the data service unit to inform the I/O processor unit of the descriptor termination. This function is performed with the use of a stack of program flags located in

the local memory portion which contains the state words and the index registers. This stack consists of consecutive state word addresses into which the program flags are stored. Each program flag indicates the number of the device which requires program service. Two countable address registers control the writing and reading of device numbers whose descriptors have been exhausted. One of the address registers contains the address at which a device number or flag can be next entered and the other address register contains the address of the device number which is to be serviced next.

If the two addresses which control the stack of program flags are not equal, the I/O processing unit is informed that the local memory unit has at least one entry in the program flag stack. Every time the I/O processor is released from a program, it looks for another program flag. The function of the I/O processor unit then is to read the terminating descriptor and check to see if the descriptor was exhausted. The program is then restarted for the device if the descriptor was exhausted and a new descriptor is constructed to direct the transfer of another block of data. This process of constructing descriptors and checking them once the descriptor is exhausted, is continued by the I/O processor unit until the job is completed, using the two buffer method, so that for every block of data which is transferred from the device to the I/O or from the I/O to the device, at least two descriptors are constructed, one for the device transfer itself and another for the data transferred to or from the disc file memory.

When a job is completed, the I/O processor sends an interrupt signal to the central processor and at the same time puts a code in the interrupt stack in the memory system. This code informs the central processor either that the job was completed properly or was not completed at all. An interrupt stack address register ISAR in the I/O processing unit retains a pointer to the interrupt stack located in a memory module. This pointer is incremented every time an entry is made in the interrupt stack. Associated with the interrupt stack pointer is an interrupt stack limit register ISLR which limits the number of entries that can be made in the interrupt stack.

Other interrupt signals may be sent to the central processor such as no access to memory module or parity error detected by the communication unit on data received from the memory module. The instructions sent to the I/O processor unit are organized in one to six syllables which are packed in memory words. One syllable is the operation code, another syllable is used for an address modifier, two syllables are used for field description, and finally two other syllables are used for addressing the memory module or the local memory. These last two syllables can be relative to either the program (instruction) base address or the channel base address. The instructions vary in length from one syllable, which is the operator, to six syllables. The reason for having variable length instructions is to achieve greater program density in the memory module.

The I/O data service unit has priority over the I/O processing unit in the local memory unit. The local memory unit is designed to accept an instruction of split cycle which allows a read cycle, then a pause, then a write cycle. This pause is taken advantage of by the data service unit to update the descriptor when a byte of information is transferred. The data service unit also has priority over the processor unit in the communication unit.

The I/O communication unit consists functionally of temporary buffer areas, a communication store register and a communication fetch register along with some timing and controls, and a parity generator and parity checker. Two four-word buffer areas are used to receive multiple word bytes coming from an input device, each having an associated address. These four-word buffer registers can also be used along with a corresponding address for a one word store in a memory module. Another address register is used to hold an address for a four-word or one-word fetch from the memory modules. Along with each address is a code which informs the memory module of the type of operation to be performed such as a one-word or four-word fetch or one-word or four-word store. A fetch instruction to the communication unit has priority over a store instruction. Ranking fourth in priority of service in the communication unit is the request from the processor unit for either a fetch or a store in memory module.

Thus the I/O module is capable of operating 512 simplex devices simultaneously. These 512 simplex devices consist of 256 output and 256 input devices or 256 half-duplex devices. The byte size and rate of data transfer of each device determine its priority in the priority scheme. Buffering all incoming and outgoing data on the disc file memory, which is a high speed device, alleviates waiting periods for the central processor.

The I/O module has the capability of handling linked descriptors such that when a descriptor is exhausted in the data service unit a new descriptor is fetched from the memory module which controls the transfer of the next block of data. The

option of linking descriptors is exercised by the I/O processor unit under program control when forming the original descriptor. This option is particularly useful in controlling transfer data from or to a real time device. The instruction to link another descriptor is placed in the control field of the originally constructed descriptor. The data service unit sets a program flag upon the termination of each descriptor, whether linked or not, to inform the I/O processor unit that the descriptor is exhausted. This permits the transfer of data to or from the memory buffer area just utilized while another buffer area is being filled or emptied in accordance with the linked descriptor. Furthermore, each linked descriptor can be linked to a further descriptor for unbroken service of an input or output device when necessary.

WHAT WE CLAIM IS:—

1. An input-output programmable computer for use with a data communication system, said computer comprising:

a processing means for constructing information transfer descriptors under program control,

an input-output controller comprising a plurality of input/output terminals and a local memory for storing address information relating to locations of input/output channel programs stored in the system and said information transfer descriptors, and

a data service means coupled to said local memory for decoding the information transfer descriptors and transferring data between said input/output terminals or between an input/output terminal and the system in response to different ones of said descriptors.

2. The apparatus according to Claim 1 wherein

said processing means is coupled to said system and to said local memory for constructing information transfer descriptors directing input/output data transfers by executing programs stored in the system which it addresses directly upon obtaining said program address information from said local memory for terminals ready to perform a new data transfer, and

said data service means is coupled to said input/output terminals and to said system for performing information transfers under control of the descriptors constructed by said processing means.

3. The apparatus according to Claim 1 or 2, wherein:

said local memory stores a channel control for each input output terminal;

said processing means further comprises means for obtaining instruction and parameter address information and means for executing the instructions utilizing the parameters so obtained for deriving new information transfer descriptors; and

said data service means further comprises means for signalling said processing means of the completion of information transfers.

4. The apparatus of Claim 1, 2 or 3, wherein

said local memory has addressable locations for storing a channel control for each input output terminal including address information of instructions and parameters stored in the system and an information transfer descripter;

said processing means is coupled to said local memory for responding to input/output channel program flags; and

said data service means further comprises means for raising individual channel program flags upon completion of information transfers specified by the descriptors.

5. The apparatus of any one of the preceding claims, wherein

said plurality of input/output terminals are coupled to peripheral input/output devices, and

said data service means further comprise priority resolving and address encoding means connected to said peripheral input/output devices and to peripheral device controllers for encoding the local memory address of the channel control of the peripheral device of highest predetermined priority which is ready to perform an information transfer job.

6. The apparatus according to any one of the preceding claims wherein said local memory further comprises a stack for storing program flags which identify specific input/output terminals ready to receive new information transfer jobs and the processing means further comprises a program flag responder for sequentially loading program flags into the local memory stack positionally in order of priority for identifying the next input/output channel program to be processed.

7. The apparatus according to any one of the preceding claims including data buffer means coupled to said input/output terminals and wherein the data service means further comprises data transfer enabling means for inserting into or selecting from said data buffer different size bytes of information for different input/output devices.

8. The apparatus according to Claim 7 wherein a portion of the local memory buffers data transfers as part of the channel controls, the descriptors of which include byte size code, byte position code, byte count, and the address of a data word storage location in the system.

9. An input-output programmable computer for use with a data communication system, said computer comprising:

a processing means for constructing in-

formation transfer descriptors under program control and for addressing a stack of input/output data transfer jobs stored in said system to obtain system address information of input output data transfer programs to be executed,

an input-output controller comprising a plurality of input output terminals and a local memory for storing address information relating to locations of input/output channel programs stored in the system of said information transfer descriptors, and

a data service means coupled to said local memory for decoding the information transfer descriptors and transferring data between said input-output terminals or between an input/output terminal and the system in response to different ones of said descriptors.

10. The apparatus according to Claim 9 wherein the system address information of input/output channel programs includes an instruction base address, an instruction count indication, and a parameter base address and the processing means comprises means for stepping said instruction count indication upon the execution of an instruction.

11. The apparatus according to any one of the preceding claims wherein certain sets of input/output program instructions stored in the system are applicable to directing information transfers by different ones of the input-output terminals, as specified by unique sets of parameters stored in the system for the input and output terminals.

12. The apparatus according to any one of the preceding claims, wherein input/output channel program routines are located in the system and a system scheduling program stored in the system assigns those routines selectively to the input/output terminals and the processing means constructs an information transfer descriptor for individual terminals for storage in the local memory, in accordance with at least one of said program routines and a set of parameters stored in the system for the individual terminal.

13. The apparatus according to Claim 1, 2 or 3 wherein

said system stores a stack of input/output data transfer jobs, and

said processing means addresses said stack responsive to the availability of an input/output terminal to perform a new information transfer job for obtaining instruction and parameter address information relating to an information transfer job to be performed by the system and stores said information in the corresponding channel control in local memory.

14. The apparatus according to any one of the preceding claims in combination with a disc file connected to selected input/output terminals, the input/output computer comprising means to transfer data between the system and the disc file directly and comprising means to transfer data between peripheral devices and the disc file utilizing the system for temporary data storage.

15. The apparatus according to any one of the preceding claims, wherein said data service means comprises means for transferring input/output data in variable size bytes responsive to the descriptors constructed by the processing means and includes priority resolution means for enabling information transfers by the input/output devices in accordance with a pre-arranged priority.

16. The apparatus according to any one of the preceding claims, wherein the program address information located in the local memory includes an instruction base address, an instruction count indication, and a parameter base address and the descriptors include byte size code, byte position code, number of bytes to be transferred, and an external memory data buffer address.

17. The apparatus according to Claim 1, 2 or 3, wherein

said local memory stores input/output channel controls including system address information of data transfer programs and information transfer descriptors.

said data service means further comprises means responsive to said descriptors for transferring data between peripheral devices coupled to said input/output terminals and said system upon demand, and

said processing means coupled to said local memory and to said system further comprises means for accessing said program address information in said local memory subject to priority access by said data service means and further comprises means for deriving said information transfer descriptors from program instructions obtained from said system.

18. An input-output programmable computer for use with a data communication system, said computer comprising:

a processing means for constructing information transfer descriptors under program control,

an input-output controller comprising a plurality of input/output terminals and a local memory for storing address information relating to locations of input/output channel programs stored in the system and said information transfer descriptors,

a data service means coupled to said local memory for decoding the information transfer descriptors and transferring data between said input/output terminals or between an input/output terminal and the system in response to different ones of said descriptors, and

a communication means for coupling said

processing means and said data service means to the system and comprising means for allowing them alternate access to said system, subject to priority for input data transfers from the data service means.

19. The apparatus according to Claim 18 wherein the means for controlling access to said system further comprises means for allowing instruction requests from the processing means to take precedence over output data transfers to the data service means.

20. The apparatus according to Claim 17 wherein the local memory includes storage locations for data words to be transferred in bytes of data and the data service means comprises means for transferring said data bytes through the input/output terminals upon demand and according to a predetermined scheme of terminal priority.

21. The apparatus according to Claim 1, 2 or 3 wherein

said local memory stores input/output channel controls relating to specific information transfer jobs,

said processing means is adapted to be coupled to the system and further comprises means for accessing stored control information relating to information transfer jobs to be performed and means for developing individual input/output channel controls from said control information; and said apparatus additionally including

a data communication means coupled to the input/output channel terminals and adapted to be connected to the system for transferring information between input/output terminals and the system main memory responsive to the channel controls stored in said addressable local memory.

22. The apparatus according to Claim 21 wherein said system has an ordered arrangement of stored input-output information transfer jobs to be performed, characterized by said processing means comprising register means for identifying the next information transfer job to be accessed and performed by said apparatus.

23. The apparatus according to Claim 21 or 22 wherein said data communication means comprises priority resolution means coupled to the input/output terminals for enabling input/output information transfers on a demand basis according to a prearranged order of priority assigned to the terminals.

24. The apparatus according to Claim 21, 22 or 23 wherein said addressable local memory comprises locations for buffering input-output information transfers such that data is transferred to and from the system in a preselected word length.

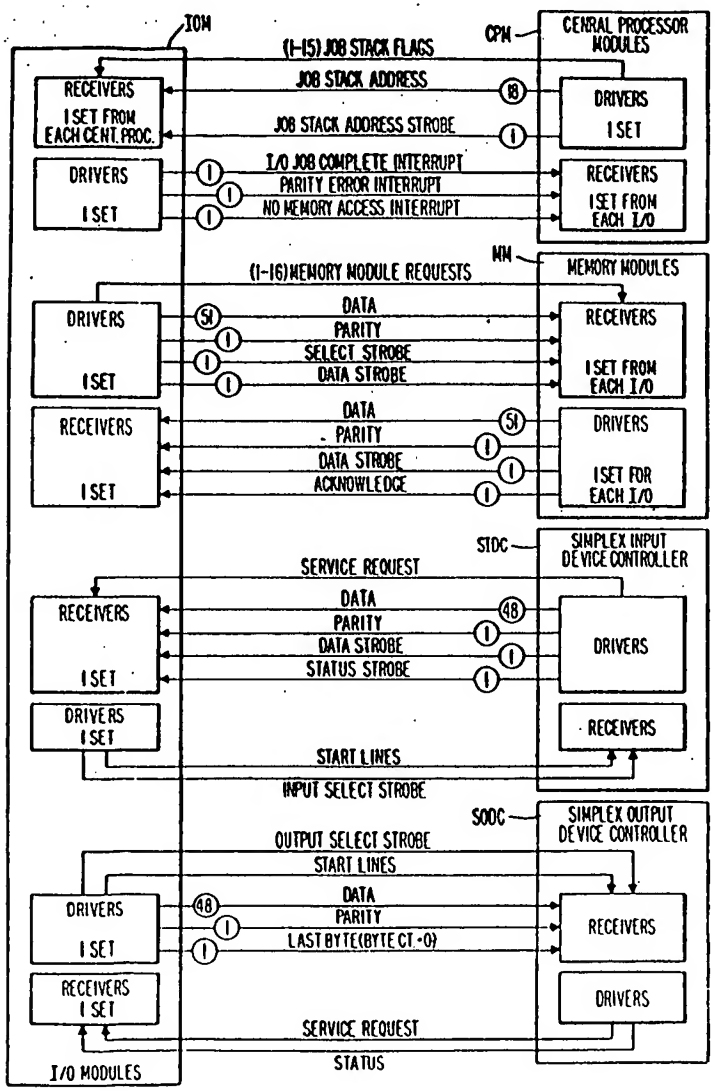
25. The apparatus according to Claim 21, 22, 23 or 24 wherein said data communication means comprises means for transferring partial data words between the addressable local memory and the input-output terminals and full data words between the system and the input-output terminals independent of said addressable local memory.

26. The apparatus substantially as herein described with reference to and as illustrated in the accompanying drawings.

BOULT, WADE & TENNANT,
111 & 112, Hatton Garden,
London, E.C.1.
Chartered Patent Agents.
Agents for the Applicants.

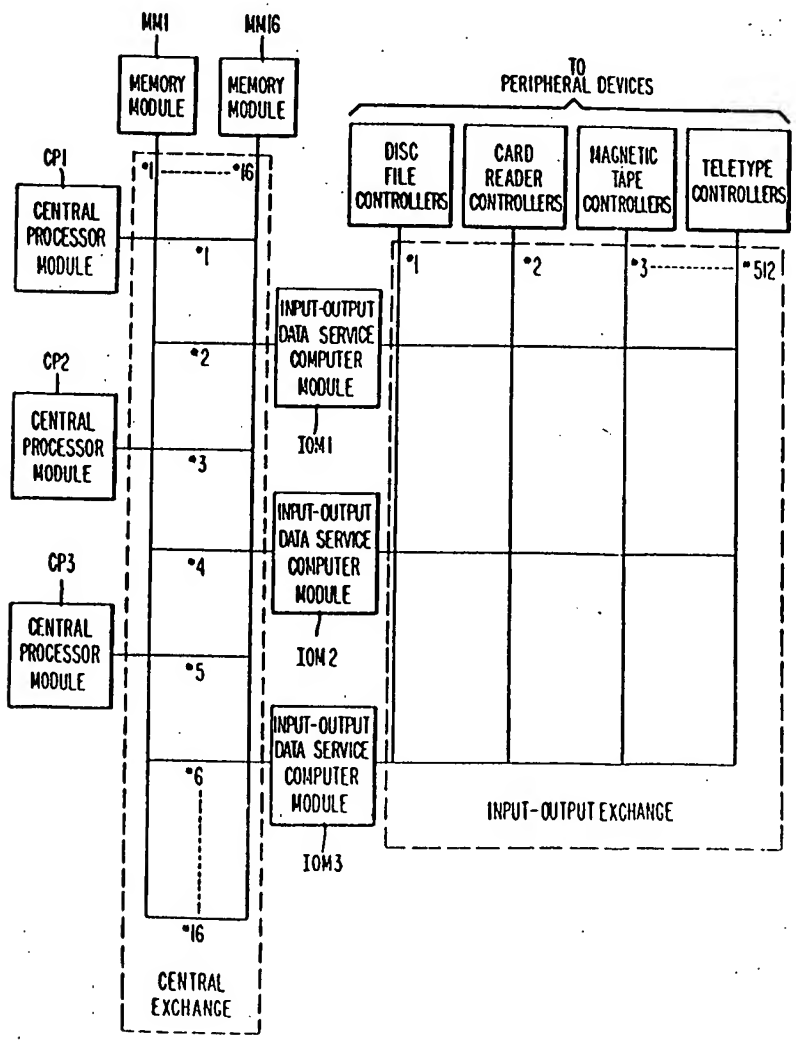
Printed for Her Majesty's Stationery Office by The Tweeddale Press Ltd., Berwick-upon-Tweed. 1969.
Published at the Patent Office, 25 Southampton Buildings, London, W.C.2, from which copies may be obtained.

Fig.13


IOM
CPM
CENRAL PROCESSOR MODULES
(1-15) JOB STACK FLAGS
JOB STACK ADDRESS
18
JOB STACK ADDRESS STROBE
1
RECEIVERS
1 SET FROM EACH CENT. PROC.
DRIVERS
1 SET
DRIVERS
1 SET
I/O JOB COMPLETE INTERRUPT
PARITY ERROR INTERRUPT
NO MEMORY ACCESS INTERRUPT
RECEIVERS
1 SET FROM EACH I/O
MM
MEMORY MODULES
(1-16) MEMORY MODULE REQUESTS
DRIVERS
1 SET
51
DATA
PARITY
SELECT STROBE
DATA STROBE
RECEIVERS
1 SET FROM EACH I/O
RECEIVERS
1 SET
DATA
PARITY
DATA STROBE
ACKNOWLEDGE
51
DRIVERS
1 SET FOR EACH I/O
SIDC
SIMPLEX INPUT DEVICE CONTROLLER
SERVICE REQUEST
RECEIVERS
1 SET
DATA
PARITY
DATA STROBE
STATUS STROBE
48
DRIVERS
DRIVERS
1 SET
RECEIVERS
START LINES
INPUT SELECT STROBE
SODC
SIMPLEX OUTPUT DEVICE CONTROLLER
OUTPUT SELECT STROBE
START LINES
DRIVERS
1 SET
DATA
PARITY
LAST BYTE (BYTE CT.=0)
48
RECEIVERS
RECEIVERS
1 SET
DRIVERS
SERVICE REQUEST
STATUS
I/O MODULES

1,167,762
23 SHEETS
COMPLETE SPECIFICATION
This drawing is a reproduction of the Original on a reduced scale.
SHEET 22

Fig. 12



STK V. EMC
STK 03051

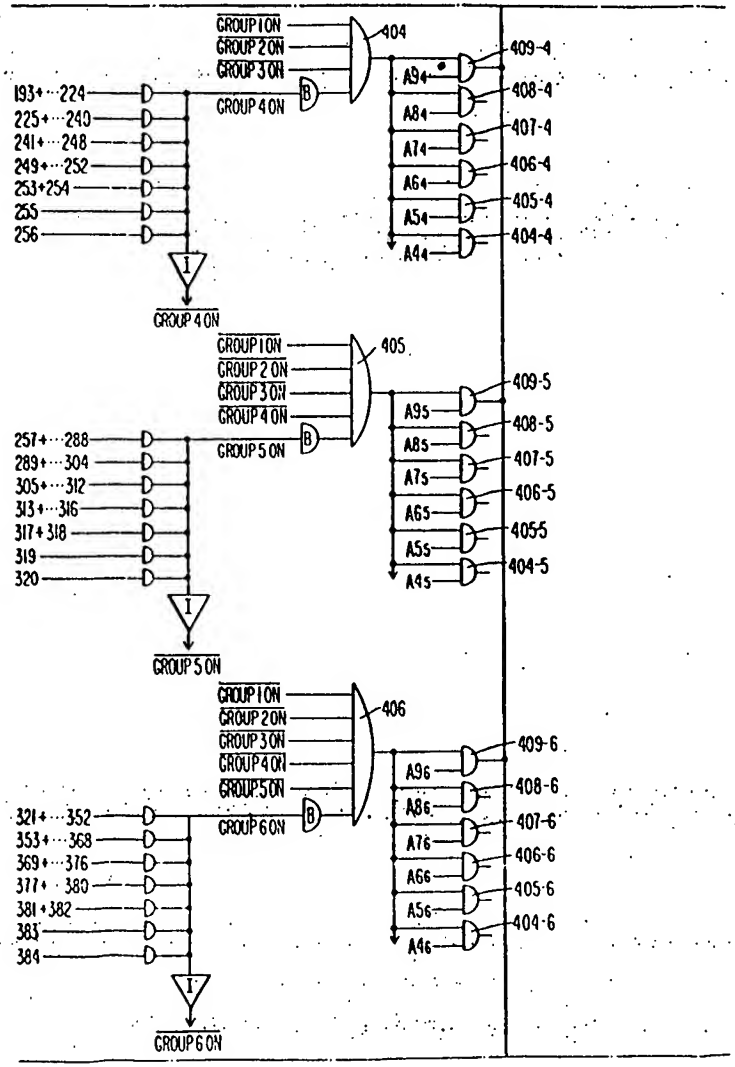

GROUP 1 ON
GROUP 2 ON
GROUP 3 ON
GROUP 4 ON
404
193+···224
225+···240
241+···248
249+···252
253+254
255
256
B
I
GROUP 4 ON
A94
A84
A74
A64
A54
A44
409-4
408-4
407-4
406-4
405-4
404-4
GROUP 1 ON
GROUP 2 ON
GROUP 3 ON
GROUP 4 ON
405
257+···288
289+···304
305+···312
313+···316
317+318
319
320
GROUP 5 ON
I
GROUP 5 ON
A95
A85
A75
A65
A55
A45
409-5
408-5
407-5
406-5
405-5
404-5
GROUP 1 ON
GROUP 2 ON
GROUP 3 ON
GROUP 4 ON
GROUP 5 ON
406
321+···352
353+···368
369+···376
377+ 380
381+382
383
384
GROUP 6 ON
I
GROUP 6 ON
A96
A86
A76
A66
A56
A46
409-6
408-6
407-6
406-6
405-6
404-6

Fig 11B

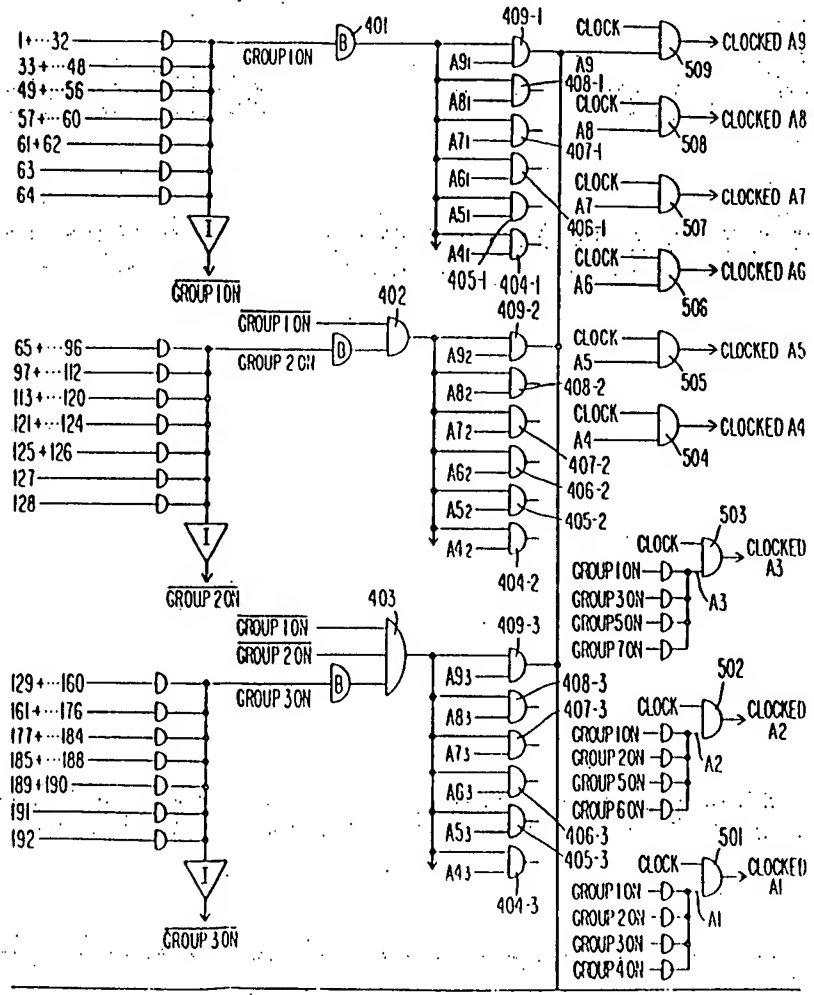

1+···32
33+···48
49+···56
57+···60
61+62
63
64
GROUP 1 ON
401
409-1
CLOCK
A9
509
CLOCKED A9
A91
408-1
A81
CLOCK
A8
508
CLOCKED A8
A71
407-1
A61
CLOCK
A7
507
CLOCKED A7
A51
406-1
A41
405-1
404-1
CLOCK
A6
506
CLOCKED A6
GROUP 1 ON
402
409-2
65+···96
97+···112
113+···120
121+···124
125+126
127
128
GROUP 2 ON
CLOCK
A5
505
CLOCKED A5
A92
408-2
A82
CLOCK
A4
504
CLOCKED A4
A72
407-2
A62
406-2
A52
405-2
A42
404-2
503
CLOCK
CLOCKED A3
GROUP 1 ON
GROUP 3 ON
GROUP 5 ON
GROUP 7 ON
A3
GROUP 1 ON
GROUP 2 ON
403
409-3
129+···160
161+···176
177+···184
185+···188
189+190
191
192
GROUP 3 ON
A93
408-3
A83
407-3
502
CLOCK
CLOCKED A2
A73
GROUP 1 ON
GROUP 2 ON
GROUP 5 ON
GROUP 6 ON
A2
A63
406-3
A53
405-3
501
CLOCK
CLOCKED A1
A43
404-3
GROUP 1 ON
GROUP 2 ON
GROUP 3 ON
GROUP 4 ON
A1

Fig. 11A

FIG.IIA
FIG.IIB
FIG.IIC

Fig.II

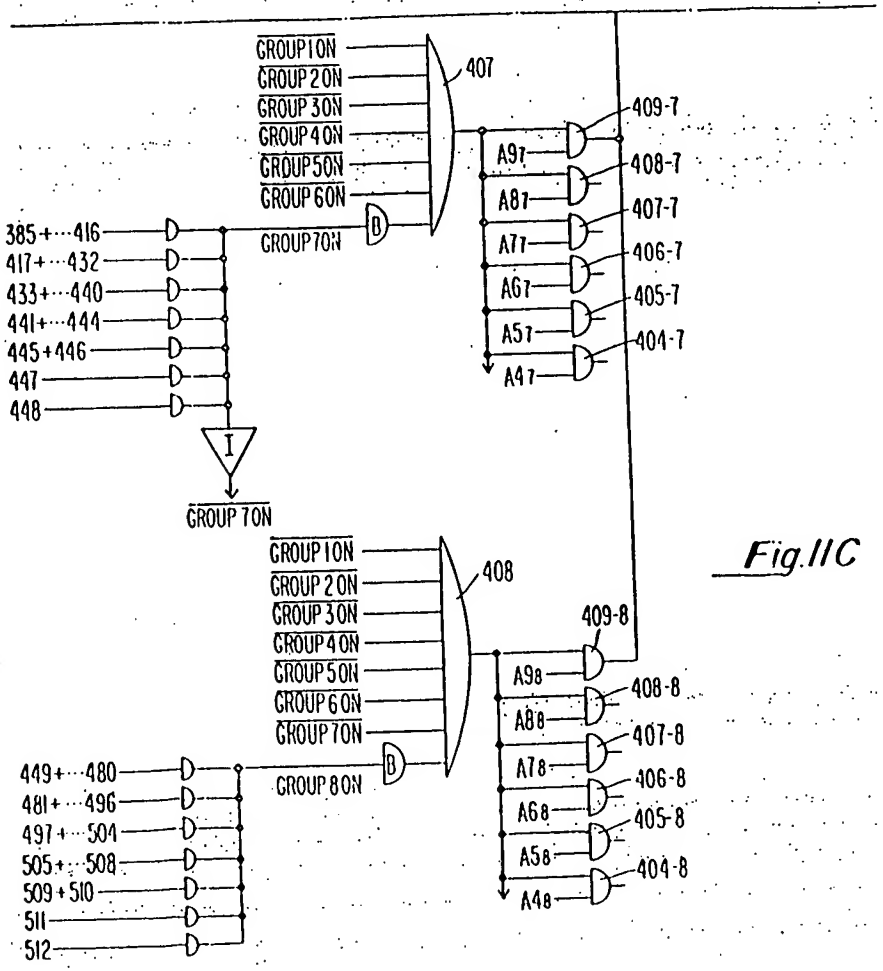


Fig.IIC

1,167,762
23 SHEETS

COMPLETE SPECIFICATION

This drawing is a reproduction of the Original on a reduced scale.

SHEET 18

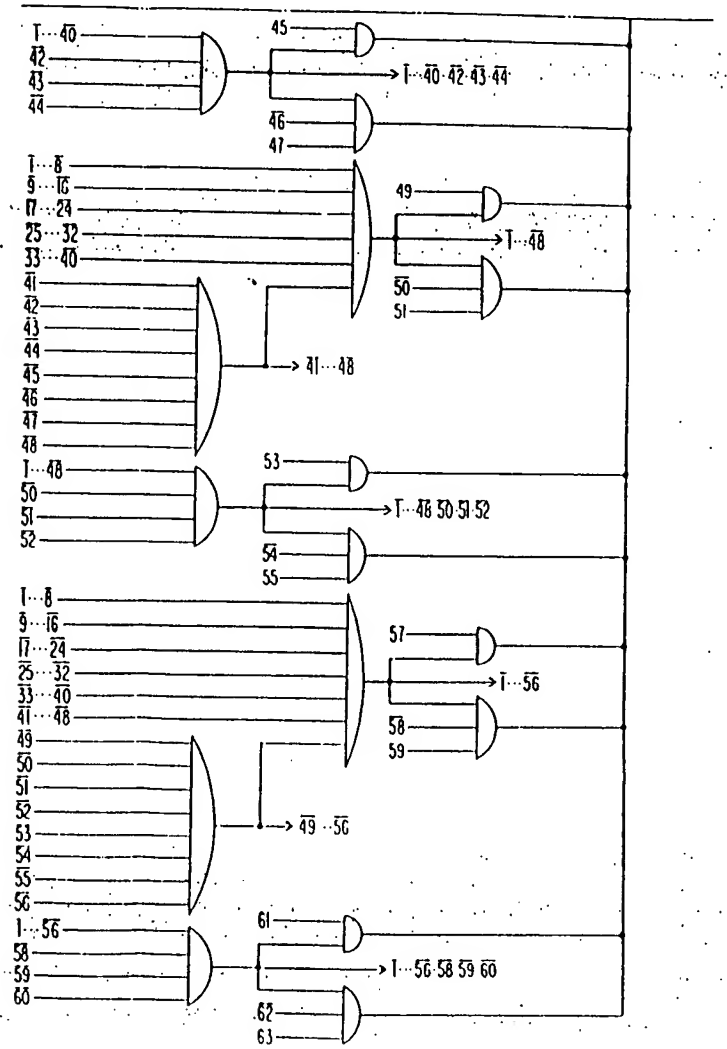


Fig 10AC

STK. 2. 1MC
SIK 03047

1,167,762
23 SHEETS
COMPLETE SPECIFICATION
This drawing is a reproduction of the Original on a reduced scale.
SHEET 17

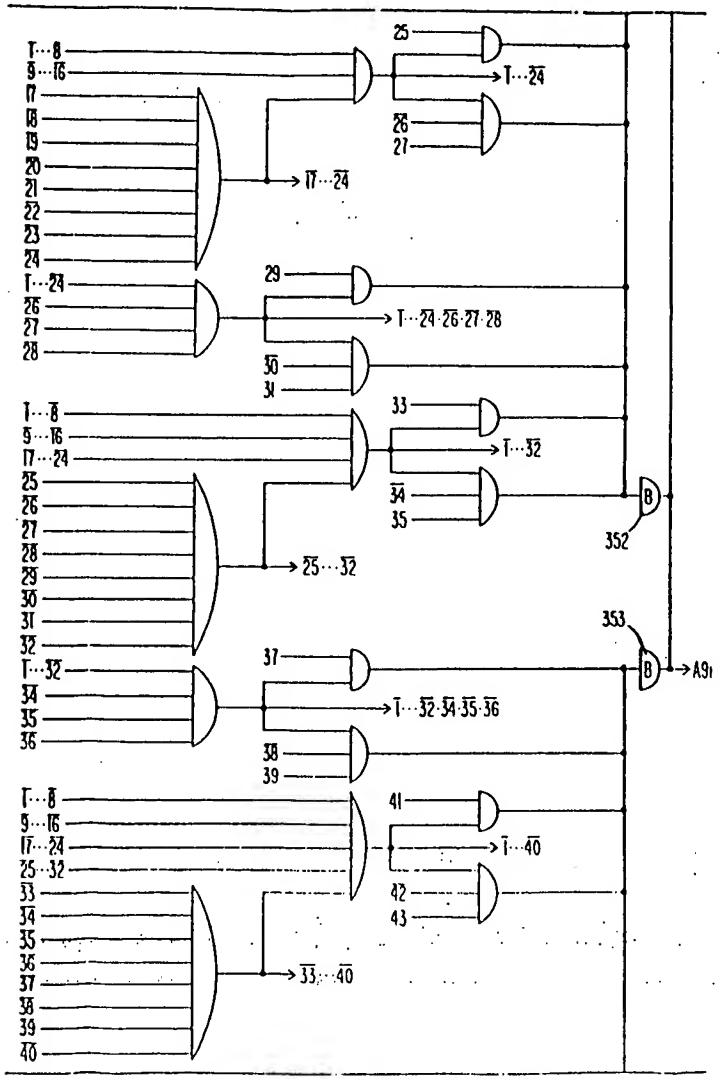


Fig.10AB

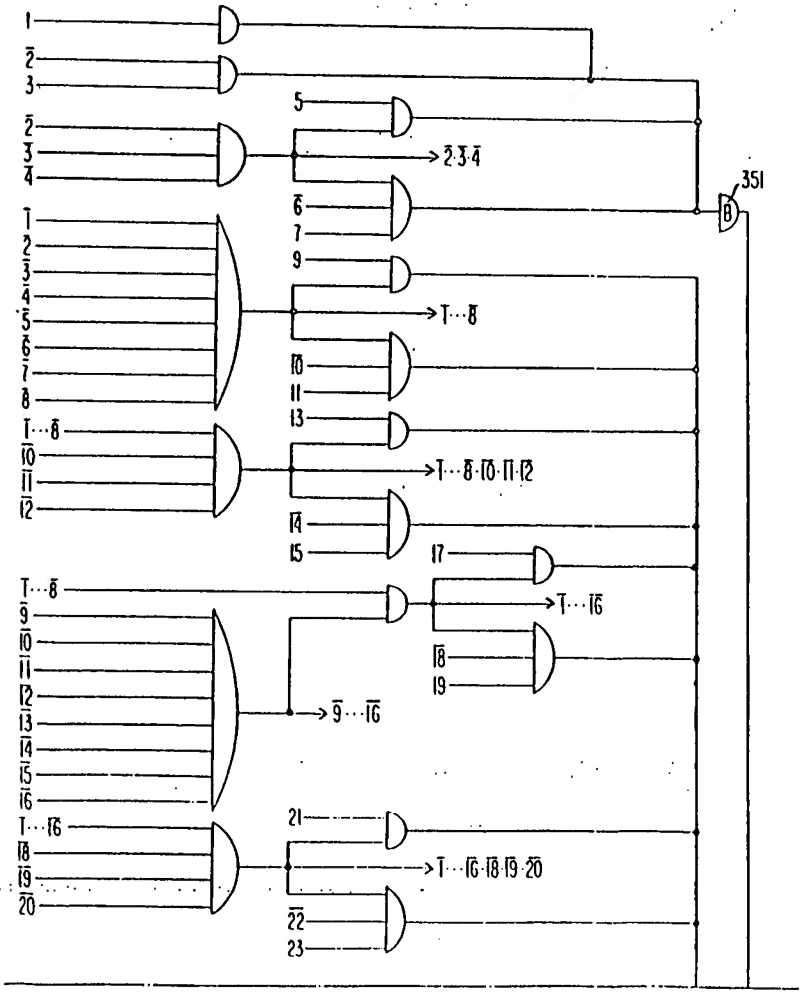

1
2̄
3
2̄
3
4
5
2·3·4̄
6̄
7
351
1
2
3
4
5
6
7
8
9
T···8
10̄
11
T···8
10̄
11
12̄
13
T···8·10·11·12̄
14̄
15
17
T···8
9
10̄
11
12̄
13
14̄
15
16̄
9̄···16̄
T···16
18̄
19
T···16
18̄
19
20̄
21
T···16·18·19·20̄
22̄
23

Fig.10AA

1,167,762 COMPLETE SPECIFICATION
23 SHEETS
This drawing is a reproduction of the Original on a reduced scale.
SHEET 15

Fig.10A

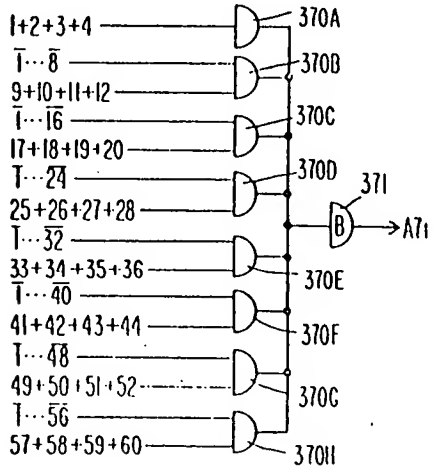


Fig.10C

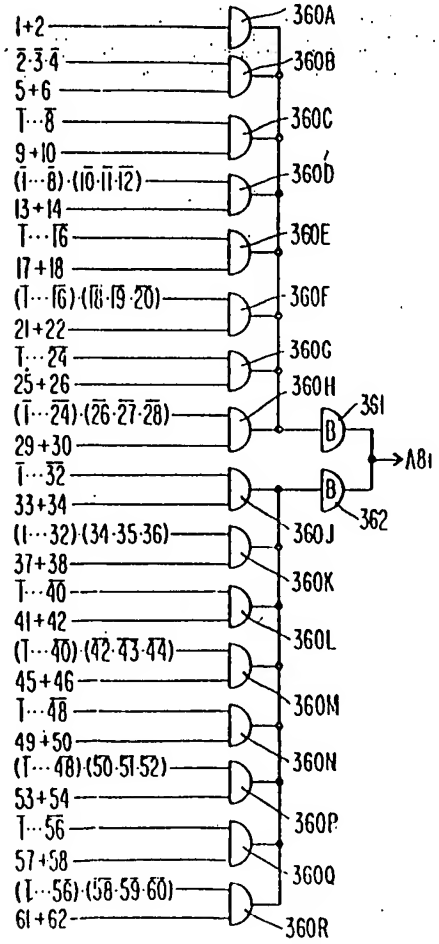


Fig.10B

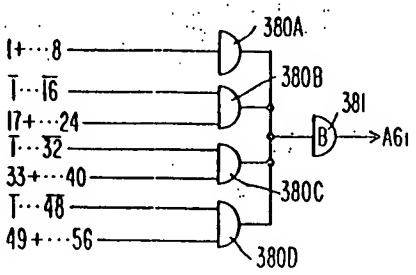


Fig.10D

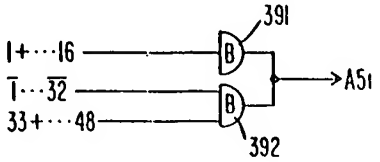


Fig.10E

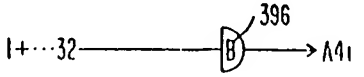


Fig.10F

1,167,762 COMPLETE SPECIFICATION
23 SHEETS This drawing is a reproduction of the Original on a reduced scale.
SHEET 14

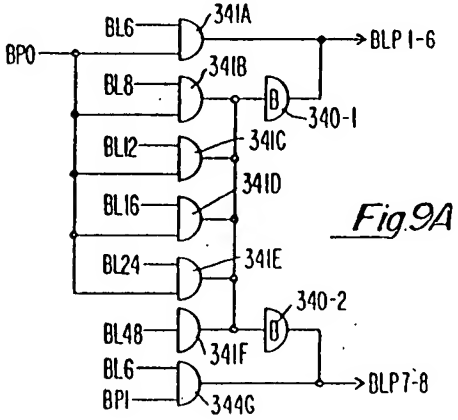


Fig.9A

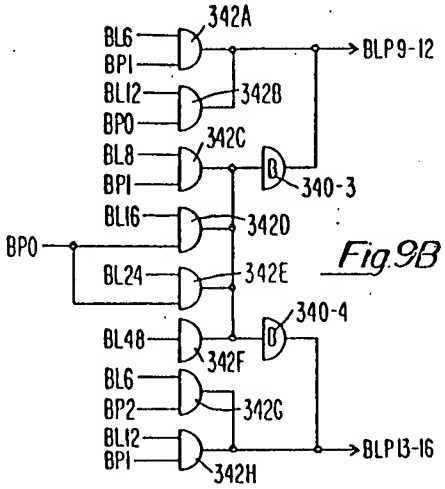


Fig.9B

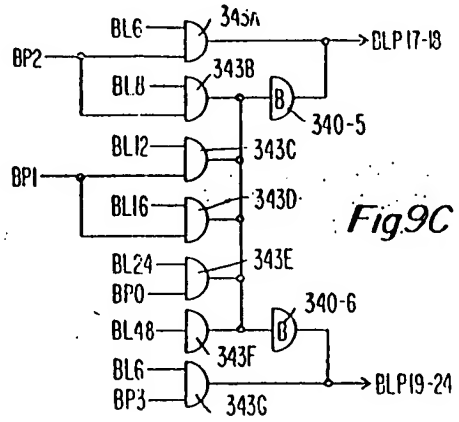


Fig.9C

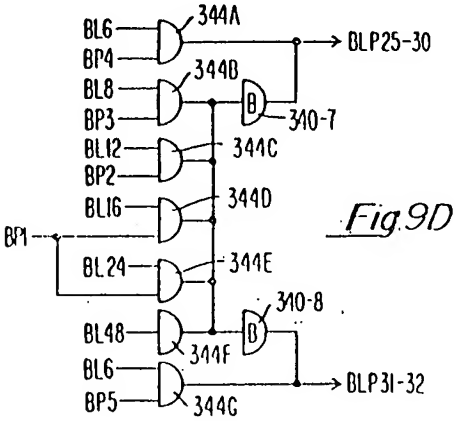


Fig.9D

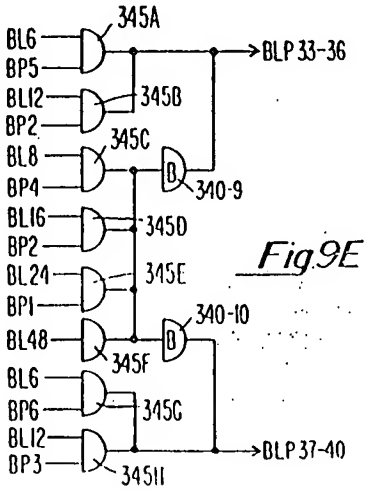


Fig.9E

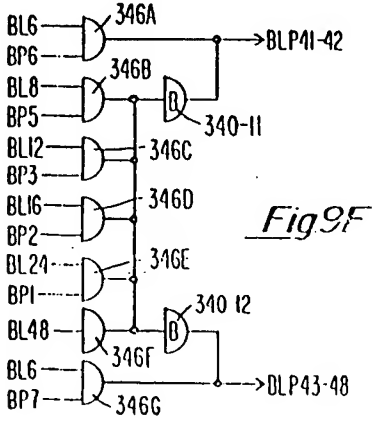


Fig.9F

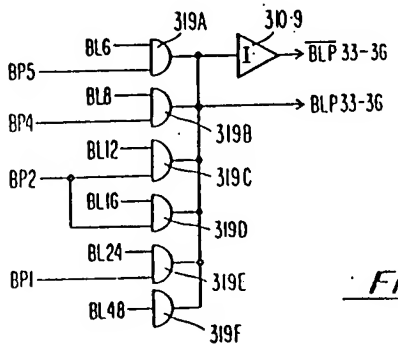


Fig.7J

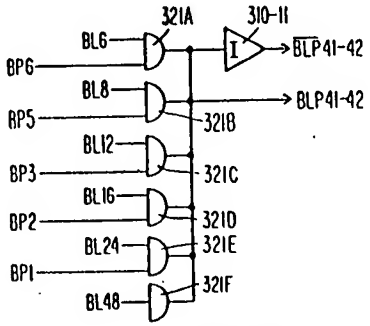


Fig.7L

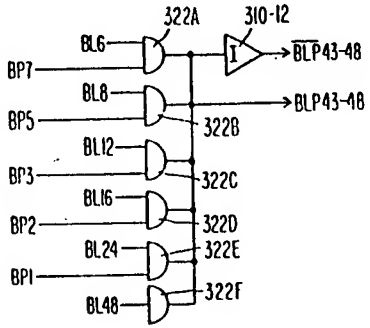


Fig.7M

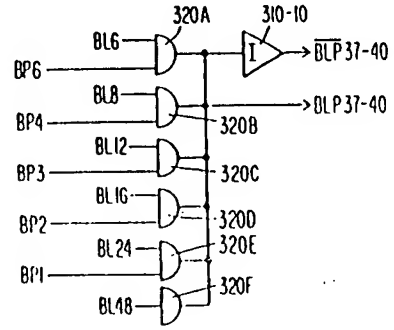


Fig.7K

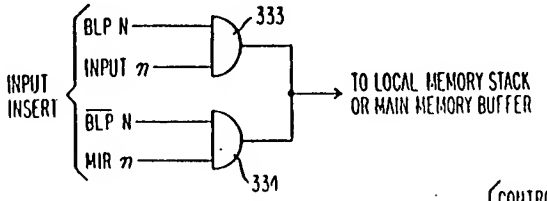


Fig.8B

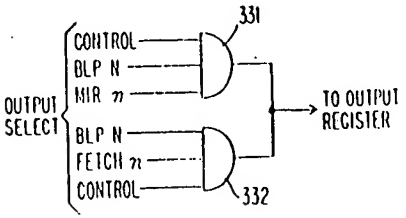


Fig.8A

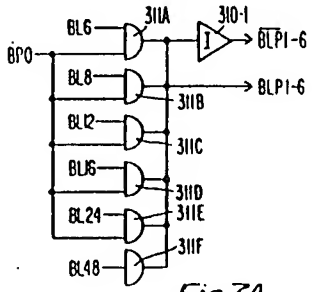


Fig.7A

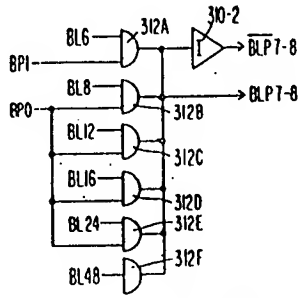


Fig.7B

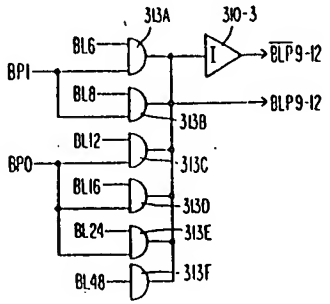


Fig.7C

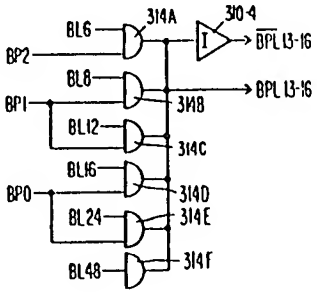


Fig.7D

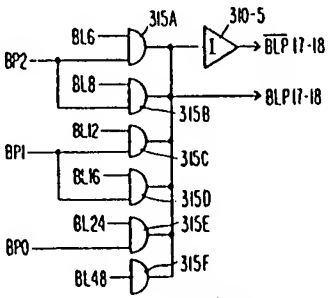


Fig.7E

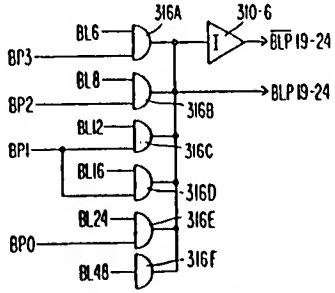


Fig.7F

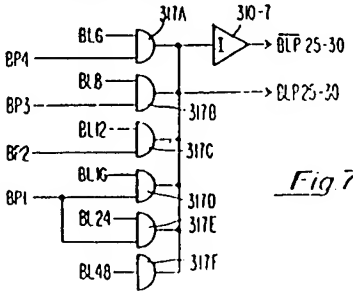


Fig.7G

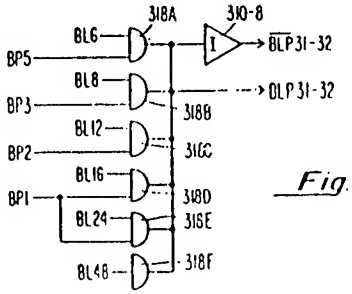


Fig.7H

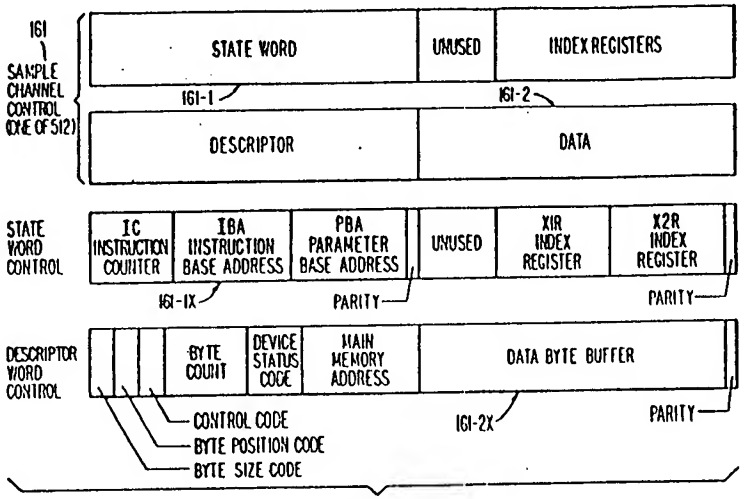


Fig.5B

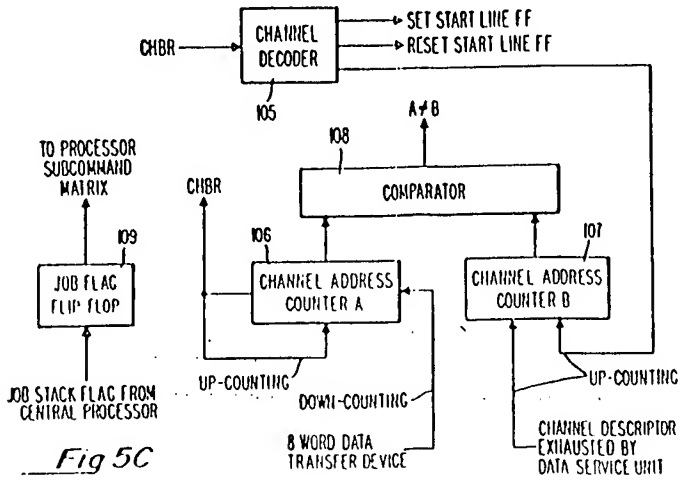


Fig 5C

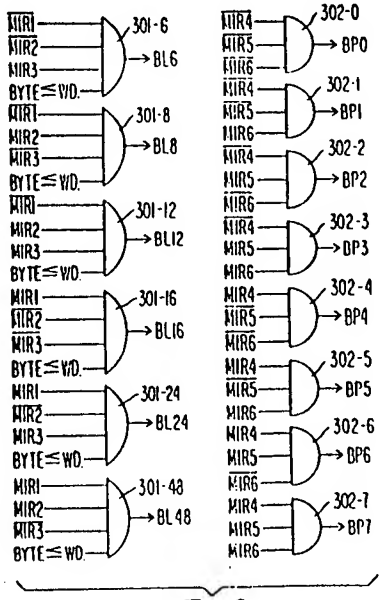


Fig.6

1,167,762
23 SHEETS
COMPLETE SPECIFICATION
This drawing is a reproduction of the Original on a reduced scale.
SHEET 10

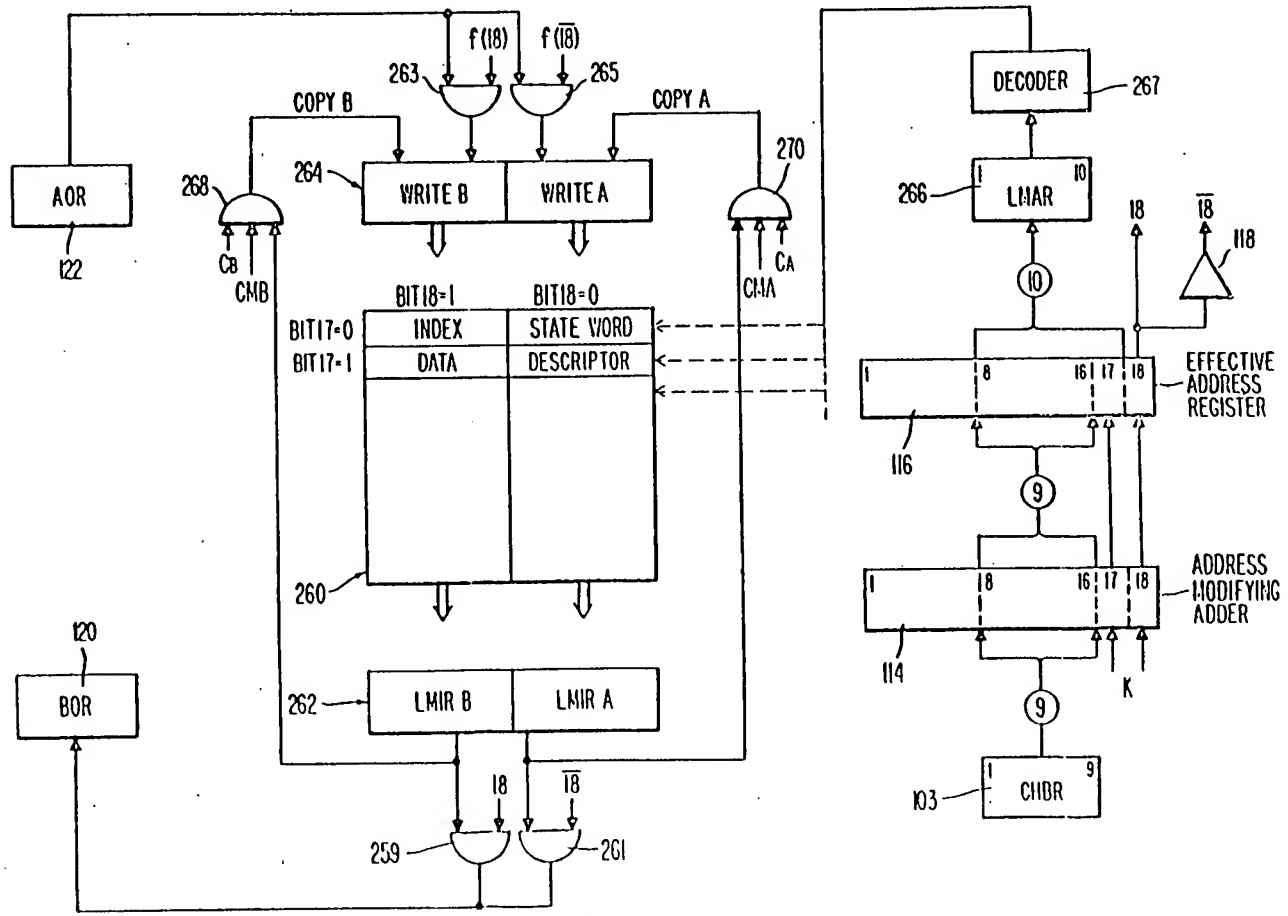
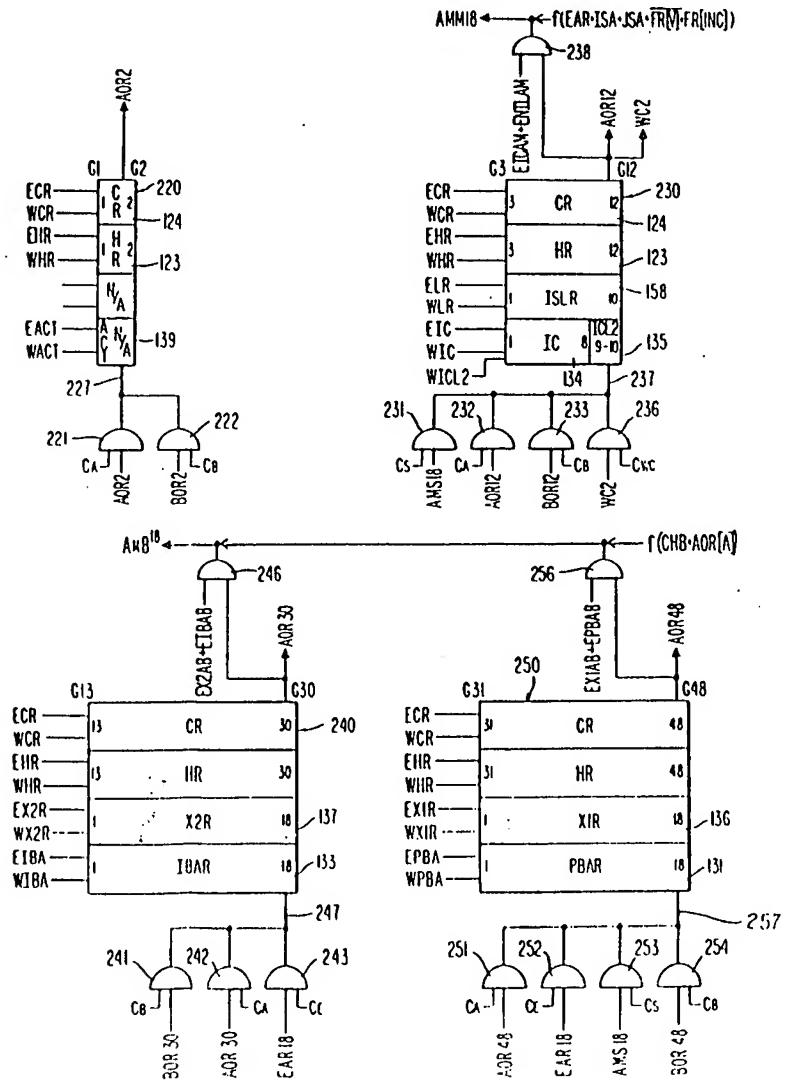


Fig.5A

1,167,762
23 SHEETS
COMPLETE SPECIFICATION
This drawing is a reproduction of the Original on a reduced scale.
SHEET 9

Fig.4



1,167,762 COMPLETE SPECIFICATION
23 SHEETS
This drawing is a reproduction of the Original on a reduced scale.
SHEET 8

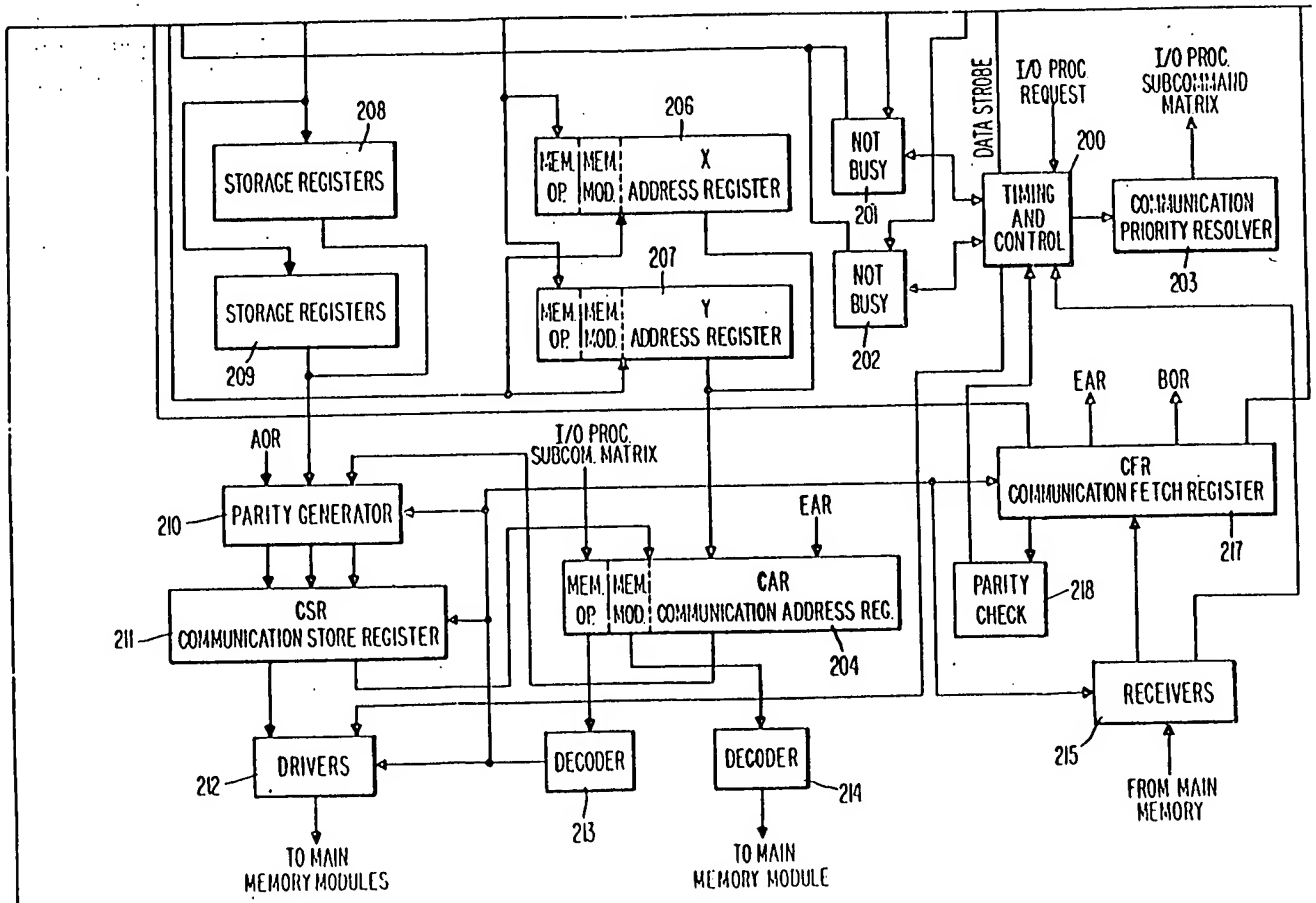


Fig.3F

1,167,762
23 SHEETS
COMPLETE SPECIFICATION
This drawing is a reproduction of the Original on a reduced scale.
SHEET 7

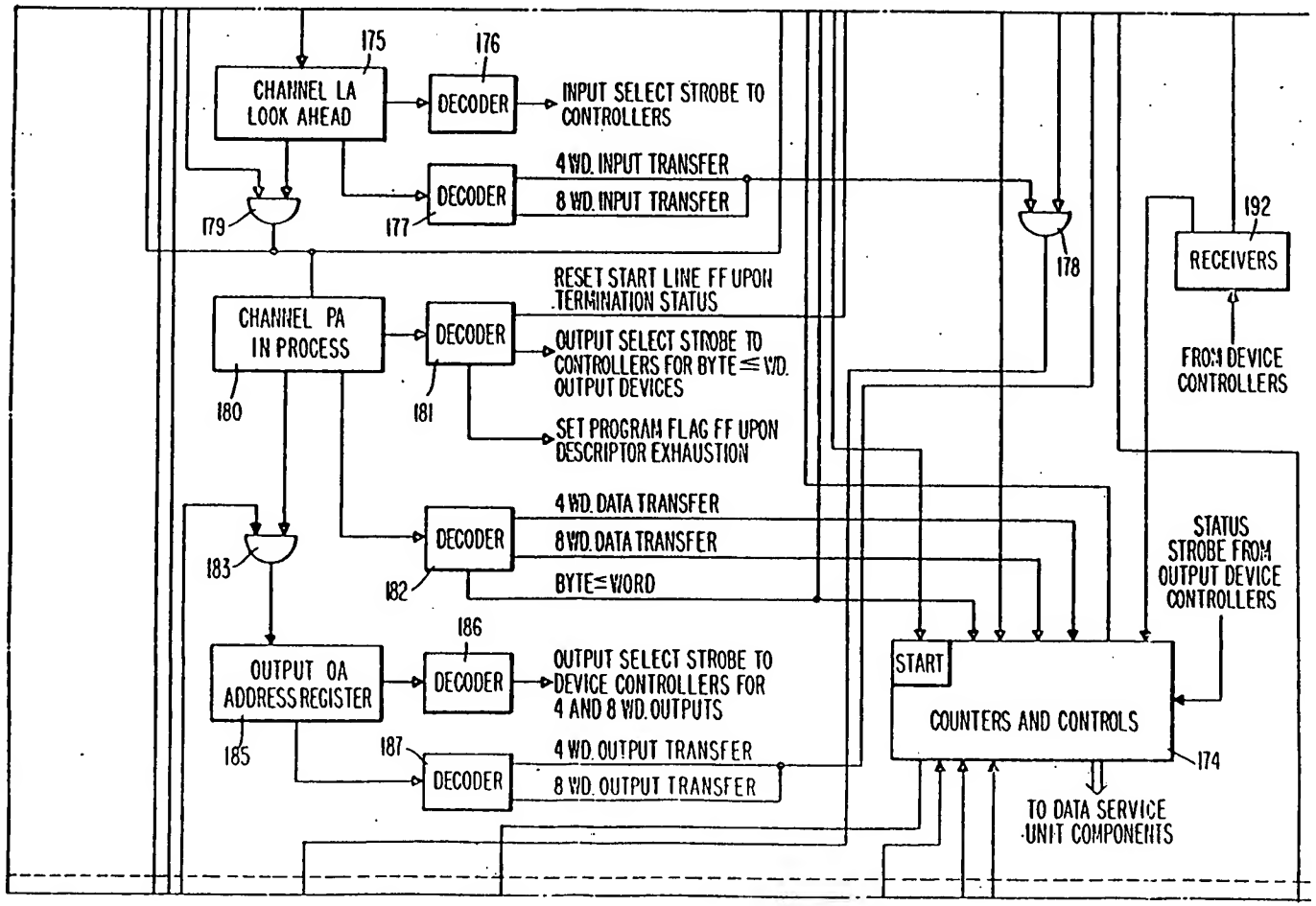


Fig 3E

STK: V. EMC
STK: 03036

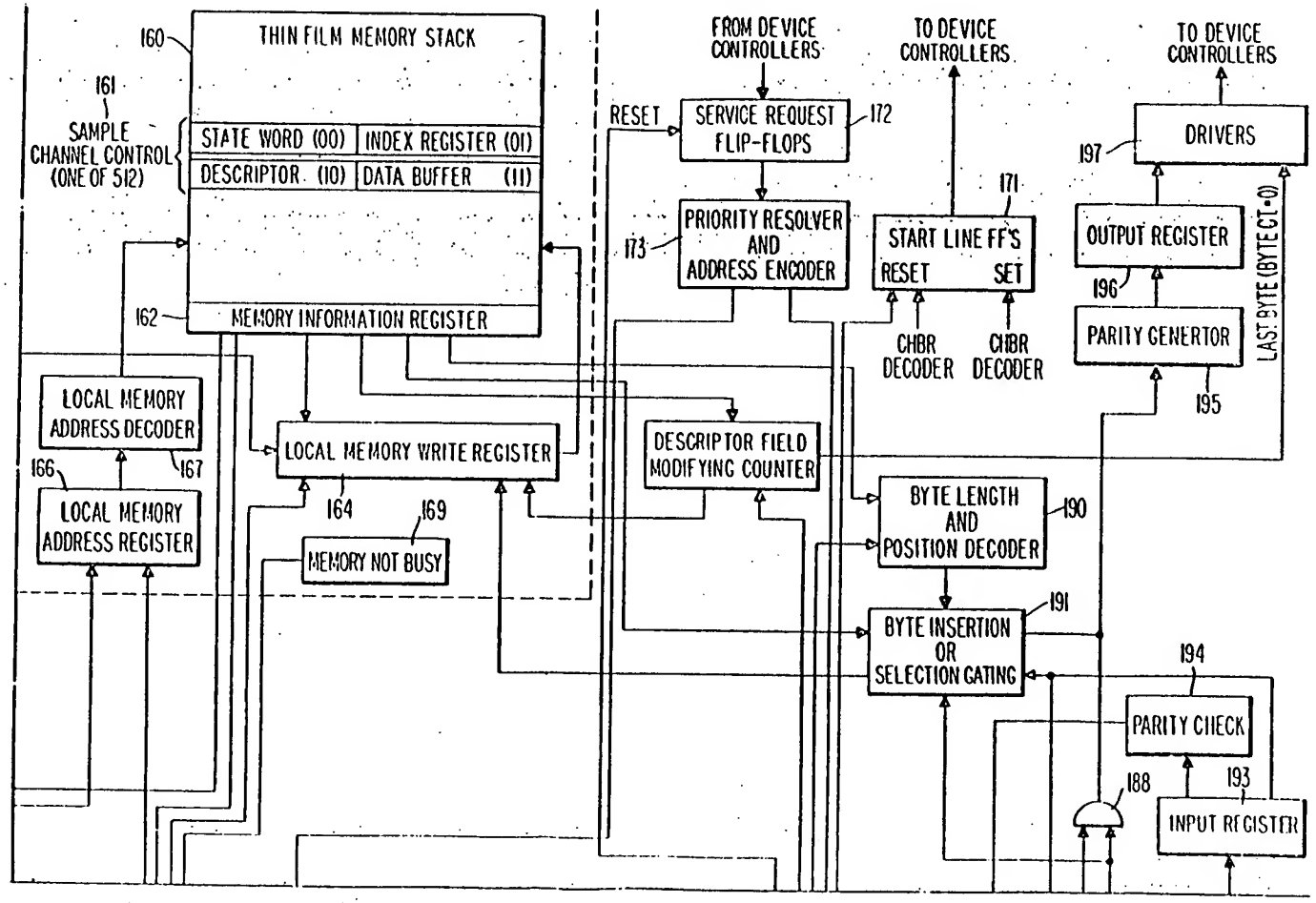

160
THIN FILM MEMORY STACK
161
SAMPLE CHANNEL CONTROL (ONE OF 512)
STATE WORD (00)
INDEX REGISTER (01)
DESCRIPTOR (10)
DATA BUFFER (11)
162
MEMORY INFORMATION REGISTER
LOCAL MEMORY ADDRESS DECODER
166
167
LOCAL MEMORY ADDRESS REGISTER
LOCAL MEMORY WRITE REGISTER
164
169
MEMORY NOT BUSY
FROM DEVICE CONTROLLERS
RESET
SERVICE REQUEST FLIP-FLOPS
172
PRIORITY RESOLVER AND ADDRESS ENCODER
173
DESCRIPTOR FIELD MODIFYING COUNTER
TO DEVICE CONTROLLERS
START LINE FF'S
RESET
SET
171
CHBR DECODER
CHBR DECODER
TO DEVICE CONTROLLERS
DRIVERS
197
OUTPUT REGISTER
196
PARITY GENERTOR
195
LAST BYTE (BYTE CT=0)
BYTE LENGTH AND POSITION DECODER
190
BYTE INSERTION OR SELECTION GATING
191
194
PARITY CHECK
188
193
INPUT REGISTER

Fig 3D

1,167,762 COMPLETE SPECIFICATION
.23 SHEETS

This drawing is a reproduction of the Original on a reduced scale.
SHEET 5

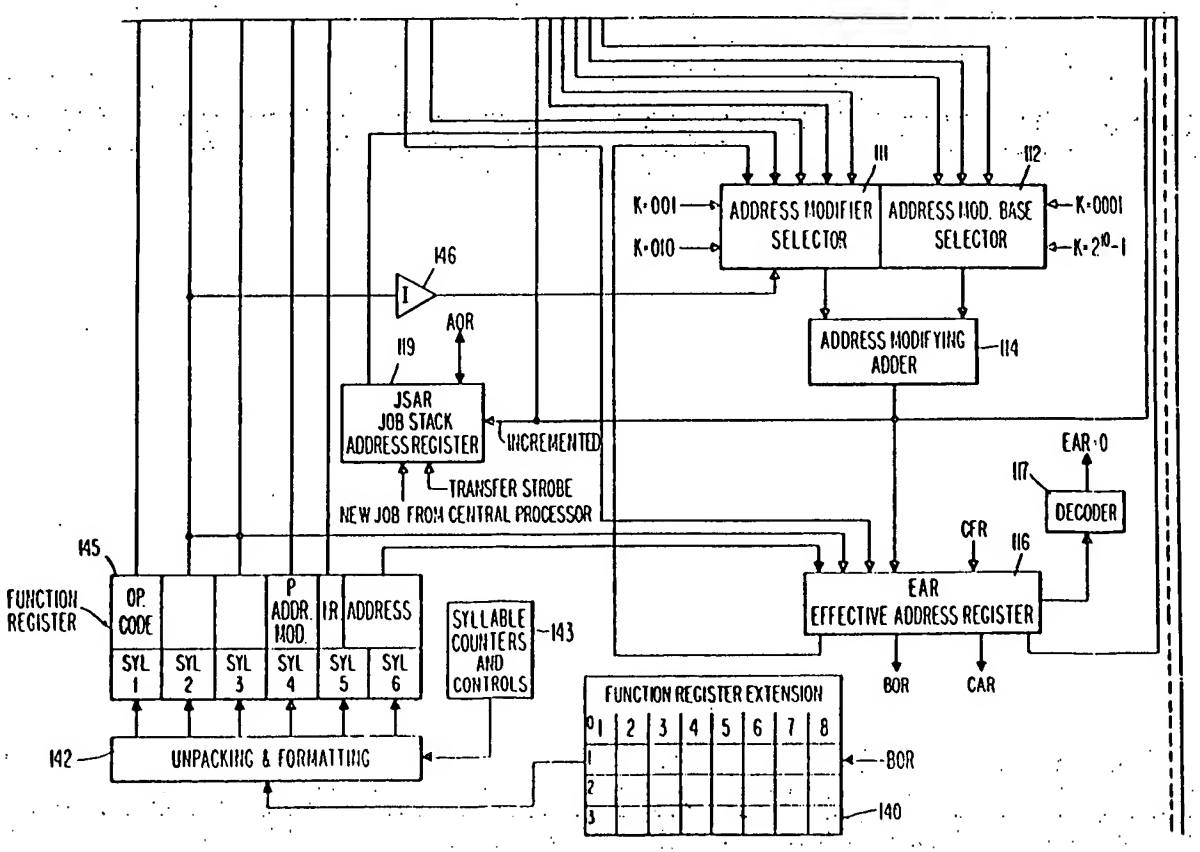


Fig 3C

1,167,762 COMPLETE SPECIFICATION
23 SHEETS
This drawing is a reproduction of the Original on a reduced scale.
SHEET 4

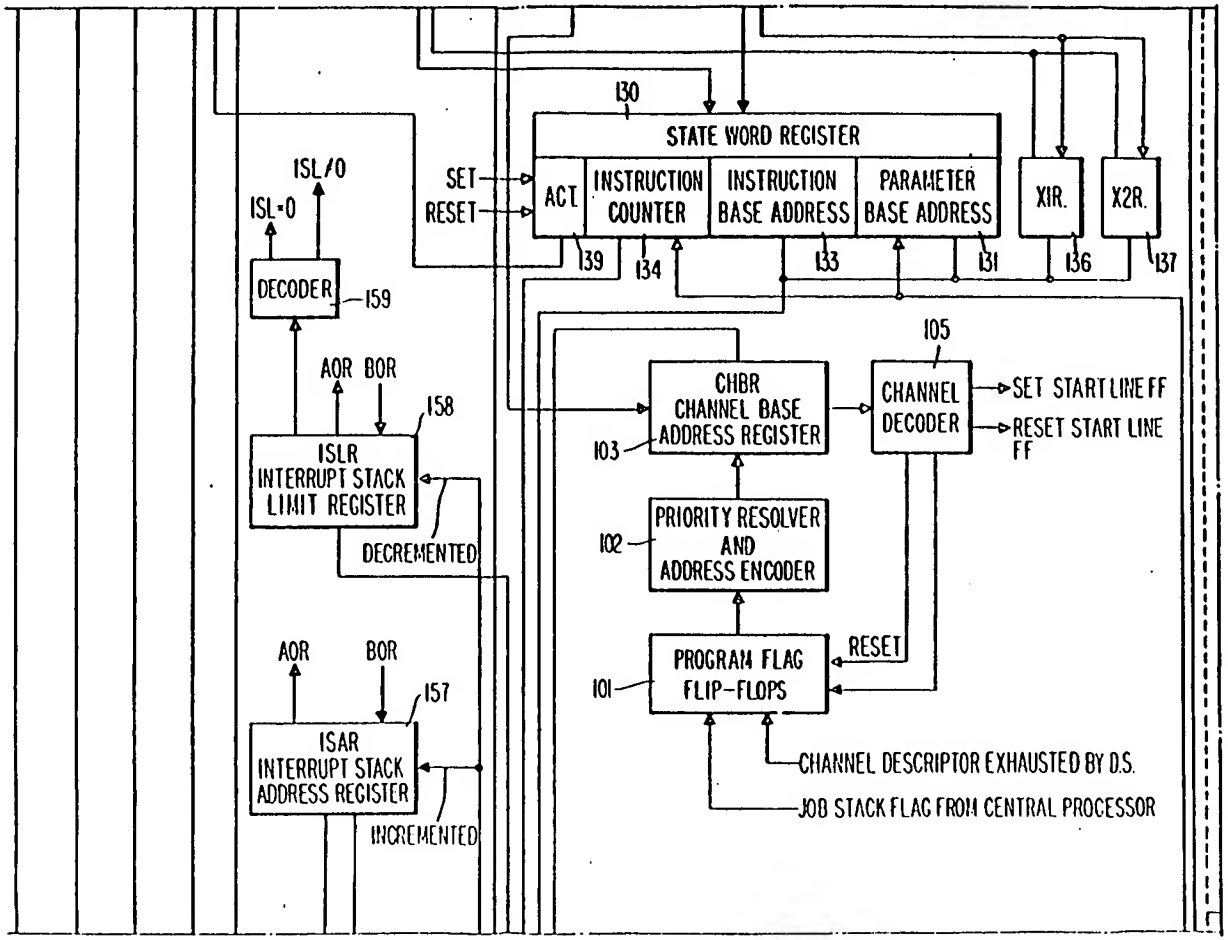


Fig 3B

1,167,762
23 SHEETS
COMPLETE SPECIFICATION
This drawing is a reproduction of the Original on a reduced scale.
SHEET 3

Fig.3

FIG.3A	FIG.3D
FIG.3B	FIG.3E
FIG.3C	FIG.3F

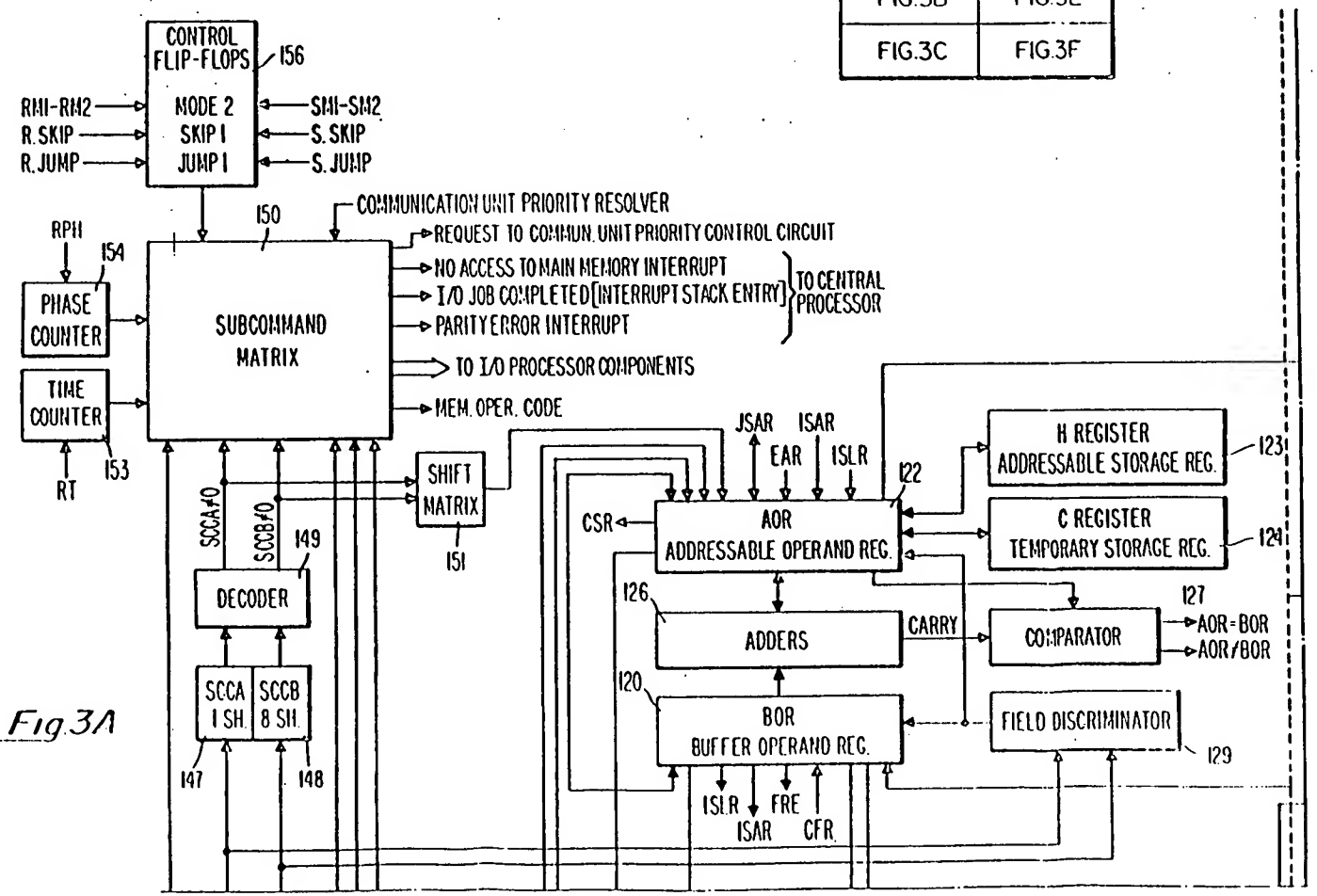


Fig.3A

1,167,762 COMPLETE SPECIFICATION
23 SHEETS

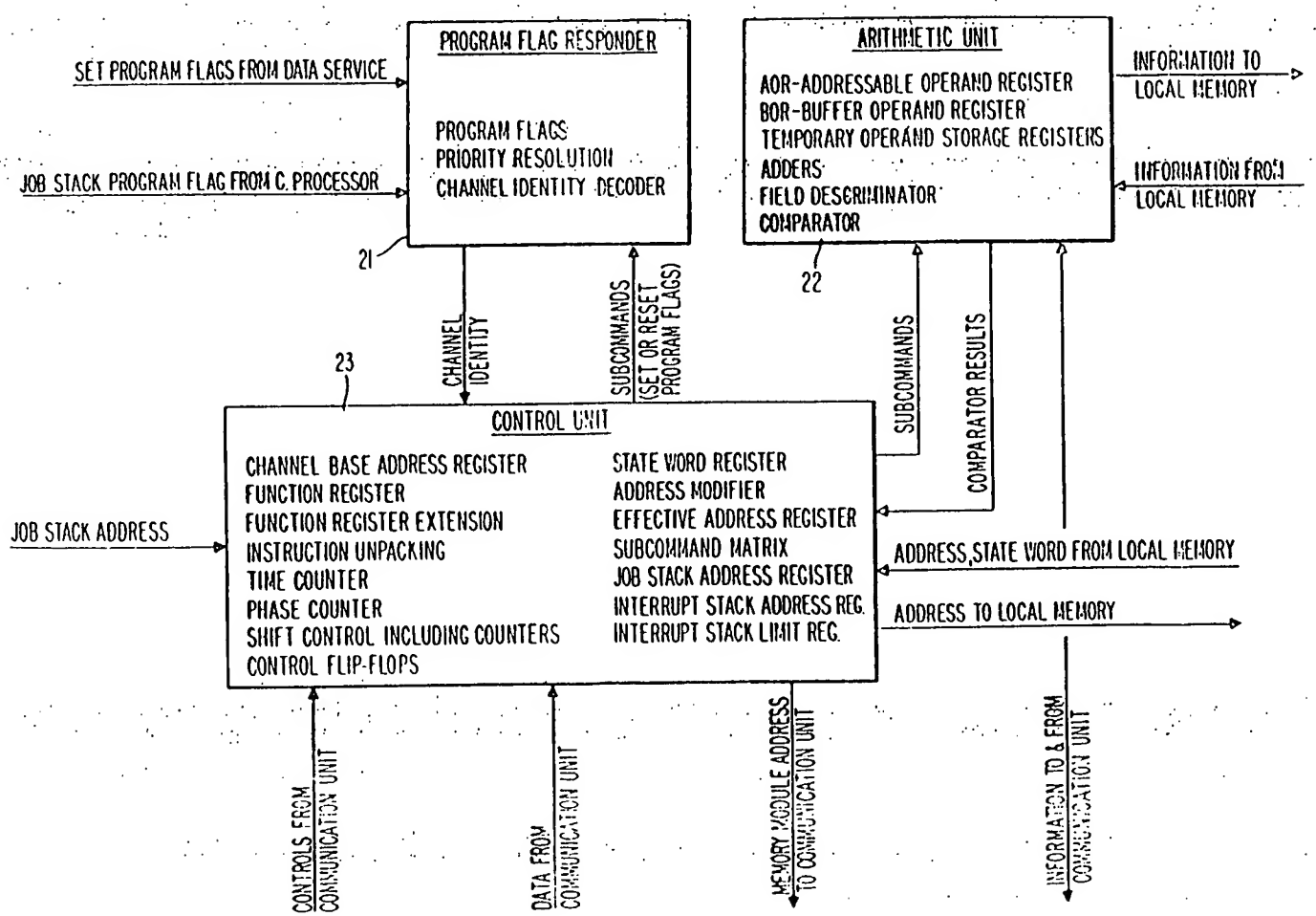


Fig 2

STK V LIM
STK 03031

1,167,762 COMPLETE SPECIFICATION
23 SHEETS

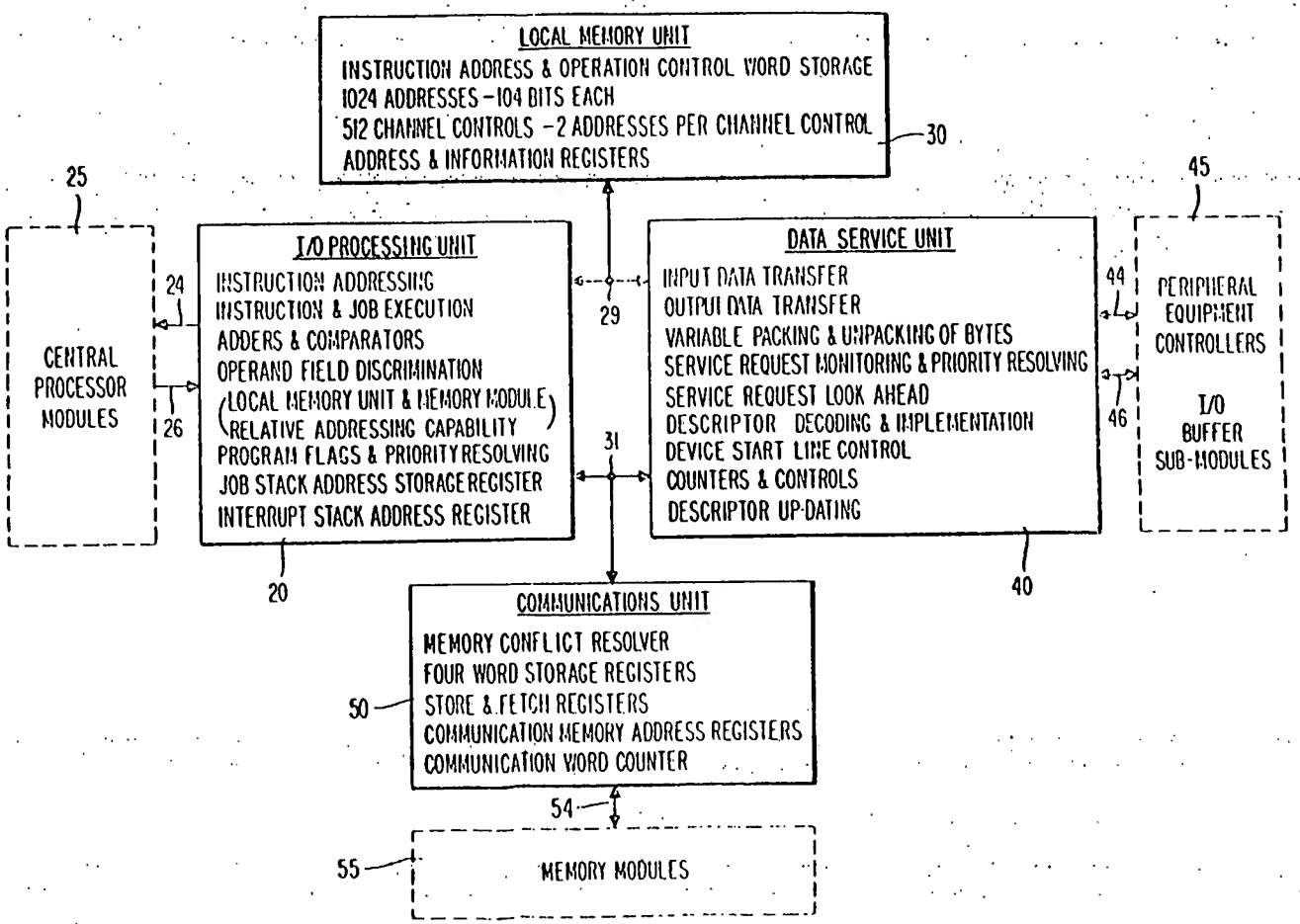


Fig 1